KOLLMORGEN

www.DanaherMotion.com

DeviceNet™

Kommunikationsprofil für

SERVOSTAR® 600





Technische Beschreibung,
Inbetriebnahmeanweisung

Ausgabe 03/04
Datei srdnet_d.xxx
EU Mat.Nr..: 104539



Bisherige Ausgaben

Ausgabe	Bemerkung
12 / 02	Erstausgabe
07 / 03	neues Layout, Objektbeschreibung geändert, diverse Korrekturen, gültig ab Firmware 5.55
03 / 04	diverse Korrekturen, auch gültig für SERVOSTAR 300 ab Firmware Version 1.0

Technische Änderungen zur Verbesserung der Geräte sind ohne vorherige Ankündigung möglich!

Gedruckt in der BRD

Seite

Seite

4 Explizite Meldungen

Seite

Seite

5 Gruppenmeldungen

6 Anhang

In diesem Handbuch verwendete Abkürzungen

Die in diesem Handbuch verwendeten Abkürzungen sind in der folgenden Tabelle erläutert.

Kürzel	Bedeutung
ACC	Beschleunigung
BOI	Kommunikationsabbruch
CAN	Controller area network
CCW	Linksdrehung
COS	Statusänderung
CW	Rechtsdrehung
EMC	Elektromagnetische Verträglichkeit
ISO	International Standardization Organization

Kürzel	Bedeutung
LED	Leuchtdiode
LSD	niederwertigste Ziffer
MAC ID	ID Medienzugriffssteuerung
M/S	Master/slave
MSD	höchstwertige Ziffer
N/A	nicht zutreffend
ODVA	Open DeviceNet Vendor Association
PC	PC mit 80x86-Prozessor

In diesem Handbuch verwendete Symbole

Symbol	Bedeutung		Symbol	Bedeutung
⚡	Gefährdung von Personen durch Elektrizität und ihre Wirkungen		⚠	allgemeine Warnung allgemeine Anweisungen mechanische Gefährdung
⇒	siehe ... (Querverweis)		●	besondere Hervorhebung

Diese Seite wurde bewusst leer gelassen.

1 Allgemeines

1.1 Über dieses Handbuch

Dieses Handbuch beschreibt die Konfiguration, den Funktionsbereich und das Softwareprotokoll der Servoverstärker SERVO**STAR**® 600 mit dem Kommunikationsprofil *Device**Net***™. Das Handbuch ist Bestandteil der vollständigen Dokumentation für die Servoverstärker der Familie SERVOSTAR.

Installation und Konfiguration des Servoverstärkers sowie alle Standardfunktionen sind in den entsprechenden Installationshandbüchern beschrieben.

Weitere Bestandteile der kompletten Dokumentation für die Reihe der digitalen Servoverstärker:

Titel	Herausgeber
Onlinehilfe der Inbetriebnahmesoftware DRIVE.EXE	Kollmorgen
Montage-/Installations-/Inbetriebnahmeanleitung SERVOSTAR	Kollmorgen

Weiterführende Dokumentation:

Titel	Herausgeber
DeviceNet Specification, Volumes I, II, Release 2.0	ODVA
CAN Specification Version 2.0	CiA e.V.
ISO 11898 ... Controller Area Network (CAN) for high-speed communication	ISO

Dieses Handbuch richtet sich an folgendes qualifiziertes Personal:



Verdrahtung: ***Fachleute mit elektrotechnischer Ausbildung***
Programmierung: ***Softwareentwickler, Projektplaner***

Wir bieten auf Anfrage Schulungs- und Einarbeitungskurse an.

1.2 Verwendung des Handbuchs

Spezifische Beispiele für einzelne Kapitel finden Sie im Anhang dieses Handbuchs.

1.3 Bestimmungsgemäße Verwendung der DeviceNet Erweiterungskarte

Beachten Sie bitte das Kapitel "Bestimmungsgemäße Verwendung" im Installationshandbuch für den Servoverstärker.

Die Schnittstelle ist ein Bestandteil der digitalen Servoverstärker der Reihe SERVOSTAR.
Die DeviceNet-Schnittstelle dient lediglich zum Anschluss des Servoverstärkers an einen Master über den DeviceNet-Bus.
Die Servoverstärker sind Komponenten für den Einbau in elektrische Geräte oder Maschinen und können nur als Einbaukomponenten solcher Geräte oder Maschinen konfiguriert und betrieben werden.



Wir können die Konformität des Servoverstärkers mit den unten aufgeführten Normen für Industriebereiche nur gewährleisten, wenn die von uns angegebenen Komponenten verwendet und die Installationsbestimmungen befolgt werden.

EMV-Richtlinie der EU ***89/336/EWG***
Niederspannungsrichtlinie der EU ***73/23/EWG***

1.4 Über DeviceNet implementierte Grundfunktionen

Bei der Arbeit mit dem in die digitalen Servoverstärker SERVOSTAR eingebauten Lageregler stehen folgende Funktionen zur Verfügung:

Konfiguration und allgemeine Funktionen:

— Referenzfahrt, Einstellen des Referenzpunkts
—- Tippbetrieb mit variabler Geschwindigkeit
—- Bereitstellung eines digitalen Sollwerts für Drehzahl- und Drehmomentregelung

Positionierungsfunktionen:

— Ausführung eines Fahrauftrags aus dem Speicher des Servoverstärkers
— Ausführung eines Direktfahrauftrags
—- absolute Trajektorie

Datenübertragungsfunktionen:

— Übertragung eines Fahrauftrags in den Speicher des Servoverstärkers
Ein Fahrauftrag besteht aus folgenden Elementen:
» Positionssollwert (absolute Aufgabe) oder Pfadsollwert (relativer Auftrag)
» Drehzahlsollwert
» Beschleunigungszeit, Bremszeit, Änderungsgeschwindigkeit/Ruckbegrenzung (i.V.)
» Art des Fahrauftrags (absolut/relativ)
» Anzahl der Folgeaufträge (mit oder ohne Pause)
—- Übertragung eines Auftrages, der keine Bewegung ist, zum Speicher des Servoverstärkers

Neben den Fahraufträgen können folgende Aufgabentypen über DeviceNet geändert werden:
— Änderung des Attributs
—- Warten bis Parameter = Wert
—- Verzweigen, wenn größer/kleiner als
—- Zähler verringern
—- Verzögerung
—- Auslesen eines Fahrauftrags aus dem Speicher des Servoverstärkers
—- Istwerte lesen
—- Fehlerregister lesen
—- Statusregister lesen
—- Konfigurations- und Steuerparameter lesen/schreiben
—- Istwerte der analogen und digitalen Eingänge lesen
—- Steuerwerte in die analogen und digitalen Ausgänge schreiben

1.5 Systemvoraussetzungen

— Servoverstärker SERVOSTAR, Ser. Nr. 730266001 oder größer oder SERVOSTAR 300
— DeviceNet-Erweiterungskarte für den SERVOSTAR
— Master-Station mit einer DeviceNet-Schnittstelle (z.B. PC mit DeviceNet-Karte)

1.6 Übertragungsgeschwindigkeit und -verfahren

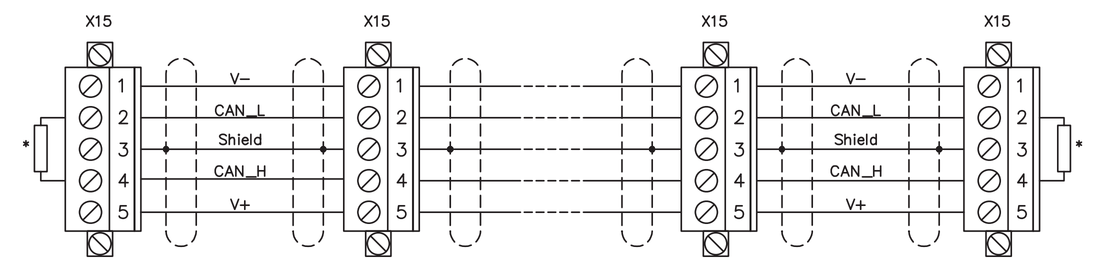
— Busanschluss und Busmedium: CAN-Standard ISO 11898 (Hochgeschwindigkeits-CAN)
— Übertragungsgeschwindigkeit: max. 500 KBit/s
Mögliche Einstellungen für den Servoverstärker: 125, 250, 500 KBit/s

1.7 Buskabel

Gemäß ISO 898 sollten Sie ein Buskabel mit einer charakteristischen Impedanz von 120Ω verwenden. Die für eine zuverlässige Kommunikation nutzbare Kabellänge wird mit ansteigender Übertragungsgeschwindigkeit reduziert. Die folgenden, von uns gemessenen Werte können als Richtlinie verwendet werden. Sie sollten jedoch nicht als Grenzwerte ausgelegt werden.

Allgemeines Merkmal	Spezifikation
Bitraten	125 KBit, 250 KBit, 500 KBit
Abstand mit dicker Sammelschiene	500 m bei 125 KBaud 250 m bei 250 KBaud 100 m bei 500 KBaud
Anzahl Knoten	64
Signalgebung	CAN
Modulation	Grundbandbreite
Medienkopplung	Gleichstromgekoppelter Differentialsende-/Empfangsbetrieb
Isolierung	500 V (optional: Optokoppler auf der Knotenseite des Transceivers)
Typische Differenzialeingangs-impedanz (rezessiver Status)	Shunt C = 5pF Shunt R = 25KΩ (power on)
Min. Differenzialeingangsimp. (rezessiver Status)	Shunt C = 24pF + 12 pF/ft der dauerhaft befestigten Abzweigleitung Shunt R = 20KΩ
Absoluter, maximaler Spannungsbereich	-25 V bis +18 V (CAN_H, CAN_L)*

* Die Spannungen an CAN_H und CAN_L sind auf den IC-Massepin des Transceivers bezogen. Diese Spannung ist um den Betrag höher als die V-Klemme, der dem Spannungsabfall an der Schottky-Diode entspricht. Diese Spannung sollte maximal 0,6 V betragen.



Erdung:
Um Erdungsschleifen zu verhindern, sollte das DeviceNet-Netzwerk nur an einer Stelle geerdet sein. Die Schaltkreise der physischen Schicht in allen Geräten sind auf das V-Bussignal bezogen. Der Anschluss zur Masse erfolgt über die Busstromversorgung. Der Stromfluss zwischen V- und Erde darf über kein anderes Gerät als über eine Stromversorgung erfolgen.

Bustopologie:
Das DeviceNet-Medium verfügt über eine lineare Bustopologie. Auf jeder Seite der Verbindungsleitung sind Abschlusswiderstände erforderlich. Abzweigleitungen bis zu je 6 m sind zulässig, sodass mindestens ein Knoten verbunden werden kann.

Abschlusswiderstände:
Für DeviceNet muss an jeder Seite der Verbindungsleitung ein Abschlusswiderstand installiert werden. Die Widerstände müssen folgende Voraussetzungen erfüllen:

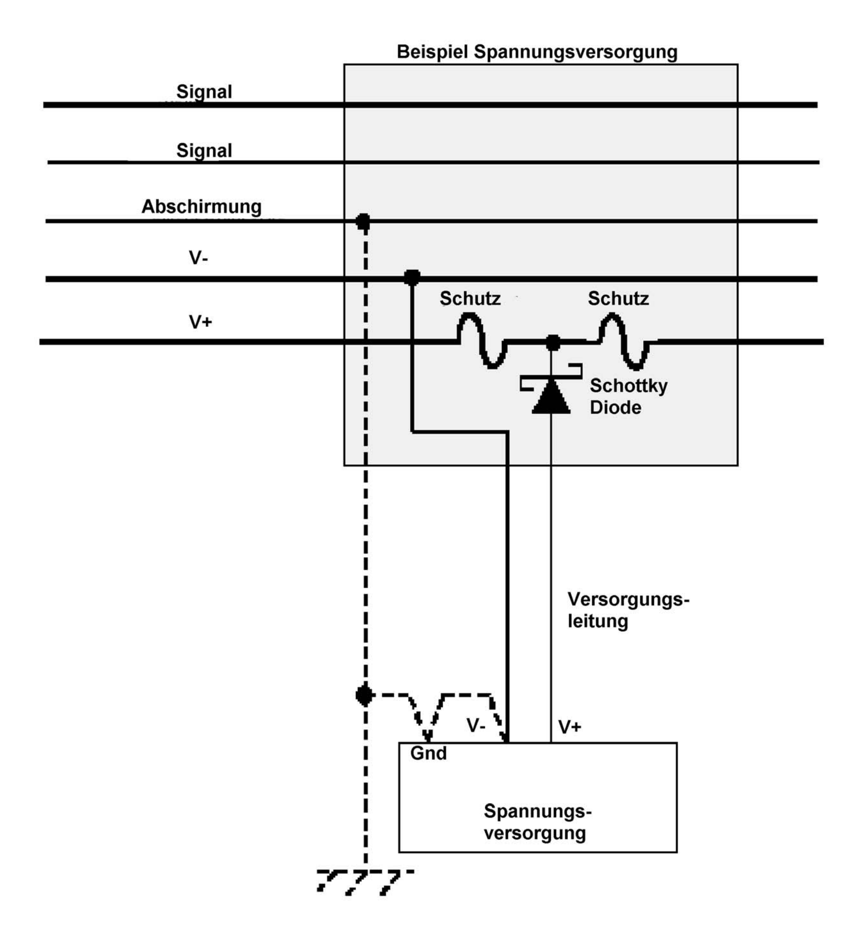
- 121Ω
- 1% Metallfolie
- 1/4 W



Wichtig: Die Abschlusswiderstände sollten nicht am Ende einer Abzweigleitung, sondern nur an den beiden Seiten einer Verbindungsleitung installiert werden.

Stromversorgung des Netzwerks:
Die Spannungsversorgungen für DeviceNet sollten folgende Eigenschaften aufweisen:

- Spezifizierte Nenndaten für Stromversorgung und Netzwerkströme (24 V)
- Sicherungen oder Leistungsschalter zur Begrenzung des Busstroms, falls die Strombegrenzung der Stromversorgung nicht ausreicht
- 10 Fuß maximale Kabellänge von der Stromversorgung zu Spannungsversorgung



1.8 Reaktion auf BUSOFF Kommunikationsfehler

Der Kommunikationsfehler BUSOFF(Kommunikationsabbruch) wird direkt durch Stufe 2 (CAN-Steuerung) überwacht und gemeldet. Diese Fehlermeldung kann verschiedene Ursachen haben.

Einige Beispiele dafür sind:
— Telegramme werden gesendet, obwohl kein anderer CAN-Knoten angeschlossen ist.
— CAN-Knoten haben unterschiedliche Übertragungsgeschwindigkeiten.
— Das Buskabel ist defekt.
— Ein fehlerhafter Kabelabschluss bewirkt Reflexionen im Kabel.

Das DeviceNet-Objekt (Klasse 0x03, Attribute 3 und 4) bestimmt die Reaktion auf einen Kommunikationsabbruch.

1.9 Kombinierte Modul-/Netzwerkstatus-LED

Status	**Zustand der LED**	**Bedeutung:**
Keine Stromversorgung / nicht online	aus	Das Gerät ist nicht online. - Das Gerät hat den Dup_MAC_ID-Test noch nicht abgeschlossen. - Das Gerät ist eventuell nicht eingeschaltet.
Gerät betriebsbereit UND online, angeschlossen	grün	Das Gerät läuft im normalen Zustand, ist online, und die Verbindungen sind im etablierten Zustand. - Das Gerät ist einem Master zugewiesen.
Das Gerät ist betriebsbereit UND online, aber nicht angeschlossen. Oder: Das Gerät ist online UND muss in Betrieb genommen werden.	blinkt grün	Das Gerät läuft im normalen Zustand, ist online, und die Verbindungen sind nicht im etablierten Zustand. - Das Gerät hat den Dup_MAC_ID-Test bestanden und ist online, aber die Verbindungen zu anderen Knoten sind nicht hergestellt. - Dieses Gerät ist keinem Master zugewiesen. - Fehlende, unvollständige oder falsche Konfiguration
Leichter Fehler und/oder Verbindungs-Wartezeit	blinkt rot	Behebbarer Fehler und/oder mindestens eine E/A-Verbindung befindet sich im Wartestatus.
Schwerer Fehler oder schwerwiegender Verbindungsausfall	rot	- Am Gerät ist ein nicht behebbarer Fehler aufgetreten; es muss eventuell ausgetauscht werden. - Ausgefallenes Kommunikationsgerät Das Gerät hat einen Fehler festgestellt, der die Kommunikation mit dem Netzwerk verhindert (z. B. doppelte MAC ID oder BUSOFF).

Diese Seite wurde bewusst leer gelassen.

2 Installation / Inbetriebnahme

2.1 Installation



Installieren und verdrahten Sie die Geräte nur im abgeschalteten Zustand, d. h. es darf weder die Netzspannung/Betriebsspannung noch die 24 V Hilfsspannung oder die Betriebsspannung anderer angeschlossener Geräte eingeschaltet sein. Achten Sie darauf, dass der Schaltschrank sicher abgeschaltet ist (Absperrung, Warnzeichen usw.). Die einzelnen Spannungen werden zum ersten Mal während der Inbetriebnahme eingeschaltet. Trennen Sie nie die elektrischen Verbindungen zum Servoverstärker, während dieser Spannung führt. Dies könnte die Elektronik zerstören.
Restladungen in den Kondensatoren können selbst einige Minuten nach Abschalten der Stromversorgung noch gefährliche Pegel haben. Messen Sie die Spannung am Zwischenkreis, und warten Sie, bis die Spannung unter 40 V gesunken ist.
Selbst wenn der Motor nicht läuft, können die Netz- und Steuerkabel noch Spannung führen.

- Montieren Sie den Servoverstärker gemäß der Installationsanleitung. Beachten Sie alle Sicherheitshinweise in der Installationsanleitung des Servoverstärkers. Beachten Sie alle Hinweise zur Montageposition, zu den Umgebungsbedingungen, zur Verdrahtung und zur Absicherung.
- Die Anschlüsse für Motor, Steuerungen und Netz sowie die Empfehlungen für ein EMV-konformes Systemlayout finden Sie in der Installationsanleitung für den Servoverstärker.

2.1.1 Anschlussmethoden

Stromversorgung, Motor, analoge Sollwerte, digitale Steuersignale, *DeviceNet*™-Verbindung siehe Installationsanleitung für den SERVOSTAR.

2.1.2 Einstellen der Stationsadresse

Die Stationsadresse (Geräteadresse im DeviceNet-Bus) für den Servoverstärker kann auf drei unterschiedliche Arten eingestellt werden:

- Stellen Sie die Drehschalter an der Vorderseite der Erweiterungskarte auf einen Wert zwischen 0 und 63. Jeder Schalter stellt eine Dezimalziffer dar. Um Adresse 10 für den Antrieb einzustellen, setzen Sie MSD auf 1 und LSD auf 0.
- Stellen Sie die Drehschalter an der Vorderseite der Erweiterungskarte auf einen Wert über 63. Sie können die Stationsadresse jetzt anhand der ASCII-Befehle DNMACID x, SAVE, COLDSTART einstellen, wobei "x" für die Stationsadresse steht.
- Stellen Sie die Drehschalter an der Vorderseite der Erweiterungskarte auf einen Wert über 63. Sie können die Stationsadresse jetzt über das DeviceNet-Objekt (Klasse 0x03, Attribut 1) einstellen. Dies geschieht normalerweise mit Hilfe eines DeviceNet-Inbetriebnahmewerkzeugs. Alle Antriebsparameter werden im nichtflüchtigen Speicher gesichert, wenn der Wert festgelegt wird. Der Antrieb muss nach der Änderung der Adresse erneut gestartet werden.

2.1.3 Einstellen der Übertragungsgeschwindigkeit

Sie können die DeviceNet-Übertragungsgeschwindigkeit auf drei unterschiedliche Arten einstellen, mögliche Übertragungsgeschwindigkeiten sind: 125, 250, 500 KBit/s.

- Stellen Sie den Drehschalter für die Baudrate an der Vorderseite der Erweiterungskarte auf einen Wert zwischen 0 und 2, 0 = 125 KBit/s, 1 = 250 KBit/s, 2 = 500 KBit/s.
- Stellen Sie die Drehschalter an der Vorderseite der Erweiterungskarte auf einen Wert über 2. Sie können die Baudrate jetzt anhand der Terminal-Befehle DNBAUD x, SAVE, COLDSTART einstellen, wobei "x" für 125, 250 oder 500 steht.
- Stellen Sie die Drehschalter an der Vorderseite der Erweiterungskarte auf einen Wert über 2. Sie können die Baudrate jetzt anhand des DeviceNet-Objekts (Klasse 0x03, Attribut 2) auf einen Wert zwischen 0 und 2 einstellen. Dies geschieht mit Hilfe eines DeviceNet Inbetriebnahmewerkzeugs. Alle Parameter werden im nichtflüchtigen Speicher gesichert, wenn der Wert festgelegt wird. Der Antrieb muss nach der Änderung neu gestartet werden.

2.1.4 Fehlerbehandlung

Es können verschiedene Parameter zur Steuerung der Fehlerbehandlung eingesetzt werden. Kommunikationsabbrüche werden vom Antrieb erkannt, wenn ein Problem mit dem DeviceNet Netzwerk vorliegt. Standardmäßig wird die Kommunikation, sofern möglich, automatisch zurückgesetzt. Um den Antrieb in einem abgeschalteten Zustand zu belassen, wenn Fehler in Form von Kommunikationsabbrüchen festgestellt werden, setzen Sie das BOI-Attribut des DeviceNet Objekts auf 0 (Klasse 0x03, Instanz 1, Attribut 3).
Standardmäßig gibt der Antrieb bei Kommunikationszeitüberschreitung (das Verhalten bei einer Zeitüberschreitung wird normalerweise automatisch von der SPS gesteuert) eine Knotenüberwachungswarnung n04 aus. Um die Warnung zu deaktivieren, setzen Sie den Terminal-Parameter EXTWD=0. Dieser Dienst wird vom SERVOSTAR 300 nicht unterstützt.
Um DeviceNet-Statusinformation zum Debuggen anzuzeigen, geben Sie DNDUMP im Terminal-Fenster ein.

2.2 Inbetriebnahme



Nur professionelles Personal mit umfangreichen Kenntnissen der Steuer- und Antriebstechnik darf den Servoverstärker in Betrieb nehmen.

Schritt	Beschreibung
Montage / Installation prüfen	Vergewissern Sie sich, dass alle Sicherheitshinweise in der Installationsanleitung für den Servoverstärker und in diesem Handbuch beachtet und durchgeführt wurden. Überprüfen Sie die Einstellung der Stationsadresse.
PC anschliessen, DRIVE.EXE starten	Stellen Sie die Parameter für den Servoverstärker mit Hilfe der Inbetriebnahmesoftware DRIVE.EXE ein.
Grundfunktionen in Betrieb nehmen	Starten Sie die Grundfunktionen des Servoverstärkers, und optimieren Sie die Strom- und Drehzahlsteuerungen. Dieser Teil der Konfiguration ist ausführlich im Handbuch der Inbetriebnahmesoftware beschrieben.
Parameter speichern	Sobald die Parameter optimiert sind, speichern Sie diese im Servoverstärker.
Buskommunikation in Betrieb nehmen	Voraussetzung: Das in Kapitel IV beschriebene Softwareprotokoll muss im Master implementiert sein. Stellen Sie die Stationsadresse und die Übertragungsgeschwindigkeit ein.
Test der Kommunikation	Verbinden Sie den SERVOSTAR mit einem Master-Gerät. Versuchen Sie mit Explicit Messaging (z. B. Lageregler Objektklasse 0x25, Instanz 0x01, Attribut "Zielposition 0x06), einen Parameter anzuzeigen/zu ändern.



Achtung:
Stellen Sie sicher, dass Maschinen oder Personen nicht durch eine unbeabsichtigte Bewegung des Antriebs gefährdet werden.

Schritt	Beschreibung
Lageregler in Betrieb nehmen	Konfigurieren Sie den Lageregler wie im Handbuch der Inbetriebnahmesoftware beschrieben.

3 DeviceNet Übersicht

Das SERVOSTAR DeviceNet Kommunikationsprofil folgt dem ODVA Standard Position Controller Device-Profil.

3.1 Funktionsübersicht

DeviceNet™ Gerätetyp	ODVA-Voraussetzungen Lageregler
Explicit Peer-to-Peer Messaging	N
I/O Peer-to-Peer Messaging	N
Baudraten	125, 250 und 500 kB
Polled Response Time	<10ms
Explicit Response Time	< 50ms (ausgenommen Parameterobjekt, < 500 ms)
Master/Scanner	N
Configuration Consistency Value	N
Faulted Node Recovery	J
I/O Slave Messaging	
Bit Strobe	N
Polling	J
Cyclic	N
Change-of-State (COS)	N

3.2 Verwendung von Explicit Messaging und Befehlsgruppen

Normalerweise wird Explicit Messaging für die Konfiguration des Antriebs und Abgerufene E/A zur Steuerung der Bewegung verwendet. Die meisten PLCs unterstützen beide Arten von Messaging gleichzeitig. Die in den Abschnitten 3.3 bis 3.5 beschriebenen Objekte werden über Explicit Messaging aufgerufen. In Abschnitt 4.5 wird die Verwendung von abgerufenen E/A beschrieben.

Explicit Messages ermöglichen den Zugriff auf jeweils einen einzelnen Parameterwert. Der Parameter wird durch die Angabe von Klassen-, Instanz- und Attributnummer in einer Meldung ausgewählt. Abgerufene E/A-Meldungen kombinieren viele Steuer- und Status-Bits in 8-Byte-Befehle und Antwortmeldungen. Sie sind nicht so vielseitig wie explizite Meldungen (nur bestimmte Parameter sind zugreifbar), aber es können mehrere Werte innerhalb einer Meldung geändert werden. Daher eignet sich Explicit Messaging besser für die Konfiguration.

Die meisten Antriebskonfigurationen erfolgen innerhalb des Lageregler-Objekts, das die meisten für die Bewegungssteuerung notwendigen Parameter umfasst. Modifizieren Sie Parameter in diesem Objekt, um die Betriebsart einzustellen und Bewegung zu konfigurieren. Zeigen Sie Parameter an, um die Statuswörter des Antriebs zu lesen. Eine zusätzliche Antriebskonfiguration kann über das Parameterobjekt erfolgen. Dabei handelt es sich um ein vom Hersteller definiertes Objekt, das herstellerspezifische Parameter umfasst. Alle Antriebsparameter mit einer DPR-Nummer (siehe ASCII-Referenz) von weniger als 256 kann über das Parameterobjekt aufgerufen werden.

Fahrsätze können über die Objektklasse “Befehlsblock“ vorab in den Antrieb programmiert werden. Positionierungsbewegungen, Zeitverzögerungen und Parameteränderungen können verknüpft werden, um ein im Antrieb gespeichertes Bewegungsblockprogramm zu erstellen. Sobald das gespeicherte Programm konfiguriert wird, kann es über das Objekt ”Blockfolgesteuerung" oder mit dem Blocknummernfeld Abgerufene E/A Befehlsmeldung und dem Startblockbit ausgeführt werden.

Abgerufene E/A wird für die meisten Bewegungssteuerungen verwendet. Steuer-Bits in einer Befehlsmeldung werden zur Aktivierung des Antriebs, eines kontrollierten Stopps, zum Initiieren von Bewegung und zum Initiieren von gespeicherten Bewegungsblockprogrammen verwendet. Befehlsmeldungen können darüber hinaus die Parameter für Zielposition, Zielgeschwindigkeit, Beschleunigung, Verzögerung und Drehmoment festlegen. Status-Bits in einer Antwortmeldung zeigen Fehlerstati und den allgemeinen Status des Antriebs. Antwortmeldungen können darüber hinaus die Ist-Position, die Sollposition, die Ist-Geschwindigkeit und das Drehmoment anzeigen.
Beispiele für die tatsächliche Verwendung entnehmen Sie bitte dem Anhang.

3.3 Bewegungsobjekte

Die folgeneden DeviceNet-Objekte werden verwendet, um den Verstärker zu konfighurieren und Bewegungsabläufe zu kontrollieren.

3.3.1 Objekt: Parameter

Klassencode	0x0F
Instanz-Nr.	1-255
Beschreibung	Das Parameterobjekt ermöglicht den direkten Zugriff auf die Konfigurationsparameter des Antriebs.

3.3.2 Objekt: Lageregler Überwachung

Klassencode	0x24
Instanz-Nr.	1
Beschreibung	Die Überwachung des Lagereglers bearbeitet dessen Fehler sowie die Referenzfahrteingänge.

3.3.3 Objekt: Lageregler

Klassencode	0x25
Instanz-Nr.	1
Beschreibung	Über die Objektklasse Lageregler werden die Betriebsart (Drehmoment, Geschwindigkeit, Lage) und die Fahrsätze konfiguriert und Bewegungen eingeleitet.

3.3.4 Objekt: Blockfolgesteuerung

Klassencode	0x26
Instanz-Nr.	1
Beschreibung	Dieses Objekt sorgt für die Ausführung der Befehlsblöcke oder Befehlsblockketten.

3.3.5 Objekt: Befehlsblock

Klassencode	0x27
Instanz-Nr.	1 bis 255
Beschreibung	Jede Instanz des Befehlsblockobjekts definiert einen speziellen Befehl. Diese Blöcke können mit andern Blöcken zu einer Befehlsblockkette verknüpft werden.

3.4 E/A Objekte

Die folgenden deviceNet-Objekte werden verwendet, um die verstärkereigenen Ein- und Ausgänge zu überwachen.

3.4.1 Objekt: Diskreter Eingangspunkt

Klassencode	0x08
Instanz-Nr.	1-4
Beschreibung	Die Objekte des diskreten Eingangspunkts ermöglichen den Zugriff auf die vier digitalen Eingänge des Antriebs.

3.4.2 Objekt: Diskreter Ausgabepunkt

Klassencode	0x09
Instanz-Nr.	1-2
Beschreibung	Die Objekte des diskreten Ausgangspunkts ermöglichen den Zugriff auf die beiden digitalen Ausgänge des Antriebs.

3.4.3 Objekt: Analoger Eingangspunkt

Klassencode	0x0A
Instanz-Nr.	1-2
Beschreibung	Die Objekte des analogen Eingangspunkts ermöglichen den Zugriff auf die beiden analogen Eingänge des Antriebs.

3.4.4 Objekt: Analoger Ausgangspunkt

Klassencode	0x0B
Instanz-Nr.	1-2
Beschreibung	Die Objekte des analogen Ausgangspunkts ermöglichen den Zugriff auf die beiden analogen Ausgänge des Antriebs. Objekt: Lageregler Überwachung

3.5 Kommunikationsobjekte

Die folgenden DeviceNet-Objekte steuern die Kommunikation zwischen Verstärker und Steuerung. Auf sie wird üblicherweise nicht direkt durch Anwenderprogramme der Steuerung zugegriffen.

3.5.1 Objekt: Identität

Klassencode	0x01
Instanz-Nr.	1
Beschreibung	Dieses Objekt ermöglicht die Identifizierung allgemeiner Informationen zum Gerät. Das Objekt "Identität" ist in allen DeviceNet-Produkten vorhanden.

3.5.2 Objekt: Message Router

Klassencode	0x02
Instanz-Nr.	1
Beschreibung	Dieses Objekt bietet einen Meldungsanschlusspunkt, über den ein Client einer beliebigen Objektklasse oder Instanz im physischen Gerät einen Dienst zuweisen kann.

3.5.3 Objekt: DeviceNet

Klassencode	0x03
Instanz-Nr.	1
Beschreibung	Dieses Objekt liefert Konfiguration und Status eines DeviceNet-Ports. Jedes DeviceNet-Produkt unterstützt nur ein DeviceNet-Objekt pro physischem Anschluss an die DeviceNet-Kommunikationsverbindung.

3.5.4 Objekt: Gruppe

Klassencode	0x04
Instanz-Nr.	1
Beschreibung	Dieses Objekt verbindet Attribute mehrerer Objekte, damit Daten von jedem Objekt über eine einzige Verbindung gesendet oder empfangen werden können. Gruppenobjekte können zur Verbindung von Eingangs- oder Ausgangsdaten verwendet werden. Ein Eingang erzeugt Daten im Netzwerk, während ein Ausgang Daten vom Netzwerk verbraucht.

3.5.5 Objekt: Explizite Verbindung

Klassencode	0x05
Instanz-Nr.	1
Beschreibung	Dieses Objekt verwaltet die expliziten Meldungen.

3.5.6 Objekt: I/O-Verbindung

Klassencode	0x07
Instanz-Nr.	2
Beschreibung	Dieses Objekt verwaltet die E/A-Meldungen.

3.6 Firmware-Version

Dieses Handbuch gilt für Firmware-Version 5.55 oder höher. Änderungen seit Version 5.55 entnehmen Sie bitte dem Anhang “Änderungsprotokoll”.

Dieses Handbuch gilt für SERVOSTAR 300 Firmware-Version 1.0 oder höher.

3.7 Supported Services

Die DeviceNet Objekte unterstützen die folgenden Dienste:

Get_Single_Attribute (Servicecode 0x0E)

Set_Single_Attribute (Servicecode 0x10)

Reset (Servicecode 0x05, Klasse 0x01, Instanz 1, Attribut 0 oder 1, Datenlänge = 0)

Save (Servicecode 0x16, Klasse 0x0F, Instanz 0, Attribut 0, Datenlänge = 0)

Falls Sie zusätzliche Informationen benötigen, lesen Sie bitte das gesamte Dokument.

3.8 Datentypen

In der folgenden Tabelle sind die Datentypen, die Anzahl der Bits sowie der minimale und maximale Wert angegeben.

Datentyp	Anzahl Bit	Minimaler Wert	Maximaler Wert
Boolean	1	0 (falsch)	1 (wahr)
Short Integer	8	-128	127
Unsigned Short Integer	8	0	255
Integer	16	-32768	32767
Unsigned Integer	16	0	65535
Double Integer	32	-2^{31}	$2^{31} - 1$
Unsigned Double Integer	32	0	$2^{32} - 1$

3.9 Sichern in den nichtflüchtigen Speicher

Antriebsparameter werden normalerweise im RAM gespeichert und nur dann im nichtflüchtigen Speicher gesichert, wenn explizit ein SAVE-Befehl eingegeben wird. Ein Speichervorgang kann über DeviceNet auf zwei Arten initiiert werden:

1) Speicherdienst des Parameter-Objekts. Senden Sie die folgende explizite Meldung:

   Service: 0x16
   Klasse: 0x0F
   Instanz: 0x00
   Attribut: 0x00
   Datenlänge: 0

2) Speicherattribut des Lageregler-Objekts. Senden Sie die folgende explizite Meldung:

   Service: 0x10
   Klasse: 0x25
   Instanz: 0x01
   Attribut: 0x65
   Datenlänge: 1
   Datenwert: 1

Diese Seite wurde bewussrt leer gelassen.

4 Explizite Meldungen

Normalerweise werden explizite Meldungen zum Konfigurieren des Antriebs und Einrichten der Antriebsparameter verwendet. Für weitere Informationen siehe Abschnitt 3.2.

4.1 Objekt "Lageregler Überwachung" (Klasse 0x24)

Die Lagereglerüberwachung bearbeitet die Fehlermeldungen des Lagereglers.

4.1.1 Fehlercodes

Der Antrieb gibt einen der folgenden Codes zurück, wenn ein Fehler während der Kommunikation über Explicit Messaging auftritt:

Aktion	Fehler	Fehlercode
Set	Attribut kann nicht eingestellt werden.	0x0E
Set oder Get	Attribut wird nicht unterstützt.	0x14
Set oder Get	Service wird nicht unterstützt.	0x08
Set oder Get	Klasse wird nicht unterstützt.	0x16
Set	Wert außerhalb des gültigen Bereichs	0x09

4.1.1.1 Konflikte des Objektstatus - 0x0C

Drei Bedingungen können dazu führen, dass die Antriebe diesen Fehlercode zurückgeben. Um fortzufahren, überprüfen Sie die Bedingung und beheben Sie sie.

Bedingung	Lösung
Bei einem Hardware- oder Softwareendschalter wird ein Befehl ausgegeben, um die Richtung des Endschalters zu ändern.	In die entgegengesetzte Richtung des Endschalters bewegen.
Ausgabe eines Befehls, der im aktuellen Modus nicht unterstützt wird (z.B. Registrierung im Geschwindigkeitsmodus)	Den Modus der Anwendung gemäß ändern oder den richtigen Befehl ausgeben.
Versuch, einen defekten Antrieb zu aktivieren	Fehler vor Aktivierung des Antriebs beheben.

4.1.2 Überwachungsattribute

Die folgenden Attribute werden in der Objektklasse Lagereglerüberwachung unterstützt. Die Instanznummer in den Klassen-, Attribut- und Instanzverknüpfungen der Lagereglerüberwachung ist immer 1.

4.1.2.1 Attribut 0x05: Allgemeiner Fehler

Beschreibung	Wenn aktiv, bedeutet dies, dass ein Antriebsfehler aufgetreten ist (Kurzschluss, Überspannung usw.). Der Fehler bezieht sich nicht auf den Eingang FAULT: Er wird zurückgesetzt, wenn die Fehlerbedingung behoben ist.		
Zugriffsregel	Get	**Vorgabe**	keine
Datentyp	Boolean	**Nichtflüchtig**	N/A
Bereich	1 = Fehlerbedingung liegt vor. 0 = Keine Fehler vorhanden.	**Siehe auch**	Fehlercode, ERRCODE (ASCII)

4.1.2.2 Attribut 0x0E: Aktive Indexebene

Dieses Attribut wird vom SERVOSTAR 300 nicht unterstützt.

Beschreibung	Mit diesem Attribut wird die aktive Ebene des Indexeingangs eingestellt.		
Zugriffsregel	Get/Set	**Vorgabe**	keine
Datentyp	Boolean	**Nichtflüchtig**	N/A
Bereich	0 = Low aktiv 1 = High aktiv	**Siehe auch**	N/A

4.1.2.3 Attribut 0x15: Registrierung aktivieren

Dieses Attribut wird vom SERVOSTAR 300 nicht unterstützt.

Beschreibung	Auf 1 setzen , um den Registrierungseingang zu aktivieren. Bei der Auslösung ist der Wert 0.		
Zugriffsregel	Get/Set	**Vorgabe**	keine
Datentyp	Boolean	**Nichtflüchtig**	N/A
Bereich	0 = Registrierung ausgelöst (Get) 1 = Registrierung aktiviert (Get/Set)	**Siehe auch**	N/A

4.1.2.4 Attribut 0x16: Registrierung Eingangsebene

Dieses Attribut wird vom SERVOSTAR 300 nicht unterstützt.

Beschreibung	Dieses Attribut gibt den Istwert des Registrierungseingangs zurück.		
Zugriffsregel	Get	**Vorgabe**	keine
Datentyp	Boolean	**Nichtflüchtig**	N/A
Bereich	0 = Low 1 = High	**Siehe auch**	N/A

4.1.2.5 Attribut 0x64: Fehlercode

Beschreibung	Liest den Fehlercode des Verstärkers aus		
Zugriffsregel	Get	**Vorgabe**	keine
Datentyp	Double Integer	**Nichtflüchtig**	nein
Bereich		**Siehe auch**	Allgemeiner Fehler, Fehler löschen, ERRCODE (ASCII)

4.1.2.6 Attribut 0x65: Fehler löschen

Beschreibung	auf 1 setzen, um Verstärkerfehler zu löschen		
Zugriffsregel	Set	**Vorgabe**	0
Datentyp	Boolean	**Nichtflüchtig**	nein
Bereich	0 = keine Aktion 1 = Feher löschen	**Siehe auch**	Allgemeiner Fehler, Fehlercode, CLRFAULT (ASCII)

4.2 Objekt Lageregler (Klasse 0x25)

Über die Objektklasse Lageregler werden die Betriebsart (Drehmoment, Geschwindigkeit, Lage) und die Fahrsätze konfiguriert und Bewegungen eingeleitet.

4.2.1 Fehlercodes

Der Antrieb gibt einen die folgenden Codes zurück, wenn ein Fehler während der Kommunikation über Explicit Messaging auftritt:

Aktion	Fehler	Fehlercode
Set	Attribut kann nicht eingestellt werden	0x0E
Set oder Get	Attribut wird nicht unterstützt.	0x14
Set oder Get	Service wird nicht unterstützt.	0x08
Set oder Get	Klasse wird nicht unterstützt.	0x16
Set	Wert außerhalb des gültigen Bereichs	0x09

4.2.1.1 Konflikte des Objektstatus – 0x0C

Drei Bedingungen können dazu führen, dass die Antriebe diesen Fehlercode zurückgeben. Um fortzufahren, überprüfen Sie die Bedingung und beheben Sie sie.

Bedingung	Lösung
Bei einem Hardware- oder Softwareendschalter wird ein Befehl ausgegeben, um die Richtung des Endschalters zu ändern.	In die entgegengesetzte Richtung des Endschalters bewegen.
Ausgabe eines Befehls, der im aktuellen Modus nicht unterstützt wird (d. h. Versuch einer Registrierung im Geschwindigkeitsmodus)	Den Modus der Anwendung gemäß ändern oder den richtigen Befehl ausgeben.
Versuch, einen defekten Antrieb zu aktivieren	Den Fehler vor Aktivierung beheben.

4.2.2 Attribute für den Lageregler

Die folgenden Attribute werden in der Objektklasse Lageregler unterstützt. Die Instanznummer in den Klassen-, Attribut- und Instanzverknüpfungen des Lagereglers ist immer 1

4.2.2.1 Attribut 0x01: Anzahl Attribute

Beschreibung	Die Gesamtzahl der Attribute, die vom Gerät in der Klasse "Lageregler" unterstützt werden.		
Zugriffsregel	Get	**Vorgabe**	
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	N/A
Bereich	N/A	**Siehe auch**	Attributliste

4.2.2.2 Attribut 0x02: Attributliste

Beschreibung	Gibt eine Liste der Attribute zurück, die vom Gerät in der Klasse "Lageregler" unterstützt werden. Die Länge dieser Liste ist in "Anzahl Attribute" festgelegt.		
Zugriffsregel	Get	**Vorgabe**	
Datentyp	Array of Unsigned Short Integer	**Nichtflüchtig**	N/A
Bereich	Die Matrixgröße wird durch Attribut 1 definiert.	**Siehe auch**	Anzahl Attribute

4.2.2.3 Attribut 0x03: Opmode

Beschreibung	wird verwendet, um den Betriebsmodus des Servoverstärkers auszulesen und einzustellen. 0=Lage (OPMODE 8). 1= Geschwindigkeit (OPMODE 0). 2=Drehmoment (OPMODE 2). Muss eingestellt werden bevor eine Bewegung eingeleitet wird.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Nein
Bereich	0 = Lagemodus 1 = Geschwindigkeitsmodus 2 = Drehmomentmodus 3 = andere (nur lesen)	**Siehe auch**	Trajektorie Start/Ende, OPMODE (ASCII)

4.2.2.4 Attribut 0x06: Zielposition

Beschreibung	Zielposition in Inkrementen. Attribut 11 (Trajektorie Start) oder das Abgerufene I/O Trajektorie starten/Daten laden Bit auf 1 setzen, um die Bewegung zu starten.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Double Integer	**Nichtflüchtig**	Nein
Bereich	-2^{31} bis 2^{31}	**Siehe auch**	Istposition, Mode-Flag inkrementell, Mode, O_P (ASCII)

4.2.2.5 Attribut 0x07: Zielgeschwindigkeit

Beschreibung	Dieses Attribut gibt die Zielgeschwindigkeit in Schritten pro Sekunde an. Verwenden Sie die Zielgeschwindigkeit für den Lagemodus und die Tippgeschwindigkeit (Attribut 22) für den Geschwindigkeitsmodus. Einheiten werden durch VUNIT festgelegt (Lageregler-Attribute 40-41).		
Zugriffsregel	Get / Set	**Vorgabe**	Gemäß Konfiguration
Datentyp	Double Integer	**Nichtflüchtig**	Ja
Bereich	Auf eine positive Zahl einstellen	**Siehe auch**	Istposition, Mode-Flag inkrementell, Mode, O_V (ASCII)

4.2.2.6 Attribut 0x08: Beschleunigung

Beschreibung	Dieses Attribut definiert im Lagemodus die Beschleunigung für die Positionierung und Referenzfahrt (ACCR) und im Geschwindigkeitsmodus die Beschleunigung für konstante Geschwindigkeit (ACC). Einheiten werden durch ACCUNIT festgelegt (Lageregler-Attribute 40-41). Alle Positionierbewegungen, die durch eine Befehlsverknüpfung oder ein Befehlsblockobjekt ausgelöst wurden, verwenden diese Beschleunigung.		
Zugriffsregel	Get / Set	**Vorgabe**	Gemäß Konfiguration
Datentyp	Double Integer	**Nichtflüchtig**	Ja
Bereich	Auf eine positive Zahl einstellen	**Siehe auch**	Verzögerung, ACC (ASCII), ACCR (ASCII)

4.2.2.7 Attribut 0x09: Verzögerung

Beschreibung	Dieses Attribut definiert im Lagemodus die Geschwindigkeitsabnahme für die Positionierung und Referenzfahrt (DECR) und im Geschwindigkeitsmodus die Geschwindigkeitsabnahme für konstante Geschwindigkeit (DEC). Einheiten werden durch ACCUNIT festgelegt (Lageregler-Attribute 40-41). Alle Positionierbewegungen, die durch eine Befehlsverknüpfung oder ein Befehlsblockobjekt ausgelöst wurden, verwenden diese Verzögerung.		
Zugriffsregel	Get / Set	**Vorgabe**	Gemäß Konfiguration
Datentyp	Double Integer	**Nichtflüchtig**	Ja
Bereich	Auf eine positive Zahl einstellen	**Siehe auch**	Beschleunigung, DEC (ASCII), DECR (ASCII)

4.2.2.8 Attribut 0x0A: Positions-Flag inkrementell

Beschreibung	Mit diesem Bit wird der Positionswert als absolut oder als relativ definiert		
Zugriffsregel	Get / Set	**Vorgabe**	1
Datentyp	Boolean	**Nichtflüchtig**	Nein
Bereich	0 = absolute Position 1 = relative Position	**Siehe auch**	Zielposition, Trajektorie Start/Ende, O_C bit 0 (ASCII)

4.2.2.9 Attribut 0x0B: Trajektorie Start/Ende

Beschreibung	Auf 1 setzen, um Trajektorie zu starten. Bleibt auf 1, bis die Bewegung abgeschlossen ist.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Boolean	**Nichtflüchtig**	Nein
Bereich	0 = Bewegungsende 1 = Trajektorie (in Bewegung) starten	**Siehe auch**	Sofortiger Stopp, kontrollierter Stopp

4.2.2.10 Attribut 0x0C: In Position

Beschreibung	Wenn dieses Flag gesetzt ist, befindet sich der Motor im Unempfindlichkeitsbereich zum Ziel.		
Zugriffsregel	Get	**Vorgabe**	1
Datentyp	Boolean	**Nichtflüchtig**	N/A
Bereich	0 = nicht in Zielposition 1 = in Position	**Siehe auch**	Trajektorie Start/Ende, INPOS (ASCII)

4.2.2.11 Attribut 0x0D: Istposition

Beschreibung	Der absolute Positionswert entspricht der Istposition in Schritten. Damit wird die Istposition neu definiert.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Double Integer	**Nichtflüchtig**	Nein
Bereich	-2^{31} bis 2^{31}	**Siehe auch**	Istposition, Mode-Flag inkrementell, PFB (ASCII)

4.2.2.12 Attribut 0x0E: Istgeschwindigkeit

Beschreibung	Dieses Attribut gibt die Istgeschwindigkeit an. Die Einheiten werden durch VUNIT festgelegt (Lageregler-Attribute 40-41).		
Zugriffsregel	Get	**Vorgabe**	0
Datentyp	Double Integer	**Nichtflüchtig**	Nein
Bereich	Positiver Messwert	**Siehe auch**	Zielgeschwindigkeit, PV (ASCII)

4.2.2.13 Attribut 0x11: Aktivieren

Beschreibung	Mit diesem Flag wird der Aktivierungsausgang gesteuert. Durch Löschen dieses Bit wird der Aktivierungsausgang deaktiviert und das zurzeit ausgeführte Bewegungsprofil abgebrochen.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Boolean	**Nichtflüchtig**	N/A
Bereich	0 = deaktivieren 1 = aktivieren	**Siehe auch**	Istposition, EN (ASCII)

4.2.2.14 Attribut 0x14: Kontrollierter Stopp

Beschreibung	Mit diesem Bit wird der Motor kontrolliert mit der zurzeit implementierten Verzögerungsrate gestoppt.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Boolean	**Nichtflüchtig**	Nein
Bereich	0 = keine Aktion 1 = kontrollierten Stopp durchführen	**Siehe auch**	Beschl., Verzögerung, Sof. Stopp, Trajektorie Start/Ende, STOP (ASCII)

4.2.2.15 Attribut 0x15: Sofortiger Stopp

Beschreibung	Mit diesem Bit wird der Motor sofort gestoppt.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Boolean	**Nichtflüchtig**	Nein
Bereich	0 = keine Aktion 1 = sofortigen Stopp durchführen	**Siehe auch**	Kont. Stopp, Trajektorie Start/Ende, DECSTOP (ASCII)

4.2.2.16 Attribut 0x16: Tippgeschwindigkeit

Beschreibung	stellt die Zielgeschwindigkeit im Geschwindigkeitsmodus ein. Mit dem Attribut "Richtung" wird die Richtung der Bewegung ausgewählt. Mit dem Attribut "Trajektoriestart" wird die Bewegung gestartet. Die Einheiten werden durch VUNIT festgelegt (Lageregler-Attribute 40-41).		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Double Integer	**Nichtflüchtig**	Ja
Bereich	Positiv	**Siehe auch**	Mode (Geschw.), Richtung, Trajecktorie Start/Ende, J (ASCII)

4.2.2.17 Attribut 0x17: Richtung

Beschreibung	Mit diesem Bit wird die Richtung des Motors im Geschwindigkeitsmodus gesteuert. Zur Erfassung der aktuellen Drehrichtung Bit auslesen		
Zugriffsregel	Get / Set	**Vorgabe**	1
Datentyp	Boolean	**Nichtflüchtig**	Nein
Bereich	0 = negative Richtung 1 = positive Richtung	**Siehe auch**	Mode (Geschw.), Referenzrichtung, J (ASCII)

4.2.2.18 Attribut 0x18: Referenzrichtung

Dieses Attribut wird vom SERVOSTAR 300 nicht unterstützt.

Beschreibung	Definiert die positive Richtung (von der Motorwelle aus gesehen).		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Boolean	**Nichtflüchtig**	Ja
Bereich	0 = positive Rechtsdrehung 1 = positive Linksdrehung	**Siehe auch**	Richtung, DIR (ASCII)

4.2.2.19 Attribut 0x19: Drehmoment

Beschreibung	Legt einen neuen Drehmomentbefehl im Drehmomentmodus fest oder liest den aktuellen Drehmomentbefehl.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Double Integer	**Nichtflüchtig**	Nein
Bereich	-3280 bis 3280 (3280 = Spitzendrehmoment)	**Siehe auch**	Mode (Drehmom.), Trajektorie Start, T (ASCII)

4.2.2.20 Attribut 0x28: Auflösung Feedback

Beschreibung	Anzahl der Istpositionsgeberschritte (oder antriebsinterne Einheiten) in einer Umdrehung (PGEARO). Die Auflösung beträgt im Allgemeinen 1048576 Schritte/Umdrehung für PRBASE = 20 oder 65536 Schritte/Umdrehung für PRBASE = 16. Die Motorauflösungs- und Sollauflösungsattribute werden zur Definition der gewünschten Auflösung von Benutzereinheiten für Geber in Bezug auf interne Einheiten verwendet. Die Geschwindigkeits- und Beschleunigungseinheiten können abhängig von den Werten von VUNIT und ACCUNIT in Bezug auf die Gebereinheiten definiert werden. Position [interne Einheiten] = Position [Benutzereinheiten] * Aufl. [Feedb.] / Aufl. [Motor] Beispiel: bei PRBASE = 20 für 2^{20} Bit/Umdr. in internen Einheiten, Aufl. feedb. auf 1048576 setzen. Für Benutzereinh. 1000 Inkr./Umdr., Aufl. Motor auf 1000 setzen.		
Zugriffsregel	Get / Set	**Vorgabe**	1048576
Datentyp	Double Integer	**Nichtflüchtig**	Ja
Bereich	positiv	**Siehe auch**	Auflösung Motor, PGEARO (ASCII), VUNIT (ASCII), ACCUNIT (ASCII), PRBASE (ASCII)

4.2.2.21 Attribute 0x29: Auflösung Motor

Beschreibung	Anzahl der benutzerdefinierten Schritte in einer Umdrehung des Motors (PGEARI). Die Motorauflösungs- und Sollauflösungsattribute werden zur Definition der gewünschten Auflösung von Benutzereinheiten für Geber in Bezug auf interne Einheiten verwendet. Die Geschwindigkeits- und Beschleunigungseinheiten können abhängig von den Werten von VUNIT und ACCUNIT in Bezug auf die Gebereinheiten definiert werden. Position [interne Einheiten] = Position [Benutzereinheiten] * Aufl. [Feedb.] / Aufl. [Motor] Beispiel: bei PRBASE = 20 für 2^{20} Bit/Umdr. in internen Einheiten, Aufl. feedb. auf 1048576 setzen. Für Benutzereinh. 1000 Inkr./Umdr., Aufl. Motor auf 1000 setzen.		
Zugriffsregel	Get / Set	**Vorgabe**	10000
Datentyp	Double Integer	**Nichtflüchtig**	Ja
Bereich	positiv	**Siehe auch**	Auflösung Feedback, PGEARI (ASCII), VUNIT (ASCII), ACCUNIT (ASCII), PRBASE (ASCII)

4.2.2.22 Attribut 0x65: Parameter speichern

Beschreibung	Auf 1 setzen, um die Antriebsparameter im nichtflüchtigen Speicher zu sichern.		
Zugriffsregel	Set	**Vorgabe**	0
Datentyp	Boolean	**Nichtflüchtig**	Nein
Bereich	0 = keine Maßnahme erforderlich 1 = Parameter speichern	**Siehe auch**	SAVE (ASCII)

4.2.2.23 Attribut 0x66: Antriebsstatus

Dieses Attribut wird vom SERVOSTAR 300 nicht unterstützt.

Beschreibung	Statuswort des Antriebs lesen. Beschreibung der Statusbit (DRVSTAT), siehe ASCII-Referenz.		
Zugriffsregel	Get	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Nein
Bereich		**Siehe auch**	DRVSTAT (ASCII)

4.2.2.24 Attribut 0x67: Trajektoriestatus

Beschreibung	Trajektoriestatuswort des Antriebs lesen. Beschreibung der Statusbit (TRJSTAT), siehe ASCII-Referenz.		
Zugriffsregel	Get	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Nein
Bereich		**Siehe auch**	TRJSTAT (ASCII)

4.3 Objekt Parameter (Klasse 0x0F)

Eine zusätzliche Antriebskonfiguration kann über das Parameterobjekt erfolgen, das einen direkten Zugriff auf die Antriebskonfigurationsparameter ermöglicht. Dabei handelt es sich um ein vom Hersteller definiertes Objekt, das herstellerspezifische Parameter umfasst.

In einer expliziten Meldung an das Parameterobjekt entspricht die Instanznummer der DPR-Nummer für den gewünschten Parameter. Diese DPR-Nummer ist in der ASCII-Befehlsreferenz zu finden. Nur die Parameter 1-254 sind über die Instanznummer zugreifbar, da sie nur ein Byte umfasst. Instanz 255 hat die Bedeutung "Attribut Parameter Nummer verwenden". Beachten sie Die Beschreibung dieses Attributes um auf Parameter über 254 zugreifen zu können.

Die Datenlänge für Set Value-Befehle können aus dem Feld “Data Type Bus/DPR” in ascii.chm festgelegt werden. Gleitkommawerte werden um 1000 skaliert, um eine ganze Zahl zu erhalten.

Antriebsprozesse (z.B. Move Home = MH, DPR/Instanz = 141) werden ausgeführt, indem ein Set Value-Befehl mit einer Datenlänge von 1 und einem Wert von 1 gesendet wird. Das Lesen des Werts oder das Einstellen des Werts auf 0 führt den Prozess noch nicht aus. Senden Sie etwa folgende explizite Meldung, um die Referenzfahrt zu initiieren: [Klasse=0x0F, Instanz=141, Attribut=0x01, Datenlänge=1, Datenwert=0x01].

4.3.1 Fehlercodes

Der Verstärker meldet einen der folgenden Fehlercodes wenn während der Kommunikation über Explicit Messaging ein Fehler auftritt.

Aktion	Fehler	Fehlercode
Set	Attribut kann nicht eingestellt werden	0x0E
Set oder Get	Attribut wird nicht unterstützt.	0x14
Set oder Get	Service wird nicht unterstützt.	0x08
Set oder Get	Klasse wird nicht unterstützt.	0x16
Set	Wert außerhalb des gültigen Bereichs	0x09

4.3.2 Parameter Attribute

Die folgenden Attribute werden von der Objektklasse Parameter unterstützt.

4.3.2.1 Attribut 0x01: Parameterwert

Beschreibung	Direkter Zugriff auf den Parameter. Datentyp und Zugriffsregel laut ASCII-Referenzdatenbank. Typen mit Fließkomma werden mit 1000 multipliziert, um einen Integerwert zu erhalten. Setzen Sie den Wert auf 1, um einen Antriebsprozess auszuführen (z.B. Referenzfahrt).		
Zugriffsregel	Abhängig vom Parameter,	**Vorgabe**	
Datentyp	Abhängig vom Parameter,	**Nichtflüchtig**	
Bereich		**siehe auch**	Zugriff, Datenlänge, ascii.chm

4.3.2.2 Attribut 0x04: Zugriff

Beschreibung	0x00 für Parameter mit Schreib-/Lesezugriff, 0x10 für Parameter mit Lesezugriff		
Zugriffsregel	Get	**Vorgabe**	
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	
Bereich		**siehe auch**	

4.3.2.3 Attribut 0x06: Datenlänge

Beschreibung	Länge des Parameters in Byte		
Zugriffsregel	Get	**Vorgabe**	
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	
Bereich		**siehe auch**	

4.3.2.4 Attribut 0x64: Parameter Nummer

Beschreibung	Nummer des Parameters auf den mit Instanz 255 zugegriffen werden soll. Um auf Parameter über 254 zuzugreifen muss die gewünschte Nummer in dieses Attribut geladen und dann die Instanz Parameter Objekt 255 verwendet werden. Dies ist ein Klassenattribut, die Parameternummer wird über Instanz 0 gesetzt. (hinzugefügt in Firmware 5.8.2)		
Zugriffsregel	Set	**Vorgabe**	0
Datentyp	Unsigned Integer	**Nichtflüchtig**	No
Bereich	1 bis grösste verwendete DPR Nummer	**siehe auch**	

Beispiel: Lesen von VLIM, DPR #290. Setzen der Parameter Nummer = 290 (Dienst=Set, Klasse=0x0F, Instanz=0x00, Attribut=0x64, Wert=0x0122). Lesen des Wertes (Dienst=Get, Klasse=0x0F, Instanz=0xFF, Attribut=0x01).

4.4 Objekt Blockfolgesteuerung (Klasse 0x26)

Dieses Objekt sorgt für die Ausführung der Befehlsblöcke oder Befehlsblockketten. Fahrsätze können über die Objektklasse “Befehlsblock“ vorab in den Antrieb programmiert werden. Diese Blocks entsprechen den SERVOSTAR Fahrsätzen. Positionierungsbewegungen, Zeitverzögerungen und Parameteränderungen können verknüpft werden, um ein im Antrieb gespeichertes Bewegungsblockprogramm zu erstellen. Sobald das gespeicherte Blockprogramm konfiguriert wird, kann es entweder über das Objekt ”Blockfolgesteuerung" oder mit dem Blocknummernfeld Abgerufene E/A Befehlsmeldung und dem Startblockbit ausgeführt werden.

4.4.1 Attribut 0x01: Block

Beschreibung	Definiert die Instanznummer des auszuführenden Anfangsblocks. Entspricht der Nummer im ASCII-Kommando MOVE [Blocknummer].		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	N/A
Bereich	1 bis 255	**Siehe auch**	Block ausführen, MOVE (ASCII)

4.4.2 Attribut 0x02: Block ausführen

Beschreibung	Führt den über Attribut 1 definierten Kommandoblock aus. Entspricht dem ASCII-Kommando MOVE		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Boolean	**Nichtflüchtig**	N/A
Bereich	0 = löschen oder durchgeführt 1 = Ausführung des Blocks	**Siehe auch**	Block, Blockfehler, MOVE (ASCII)

4.4.3 Attribut 0x03: Aktueller Block

Beschreibung	Nummer des in Ausführung befindlichen Blocks. Ist 0 während einer Referenzfahrt.		
Zugriffsregel	Get	**Vorgabe**	N/A
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	N/A
Bereich	1 bis 255	**Siehe auch**	Block, Block ausführen, TASKNUM (ASCII)

4.4.4 Attribut 0x04: Blockfehler

Beschreibung	Wird gesetzt, wenn ein Blockfehler auftritt. In diesem Fall wird die Ausführung des Blocks unterbrochen. Dieses Bit wird nach dem Lesen des Blockfehlercodes (5) zurückgesetzt.		
Zugriffsregel	Get	**Vorgabe**	0
Datentyp	Boolean	**Nichtflüchtig**	
Bereich	0 = Keine Fehler 1 = Blockfehler aufgetreten	**Siehe auch**	Block ausführen, Blockfehlercode

4.4.5 Attribut 0x05: Blockfehlercode

Beschreibung	Spezifischer Blockfehler, wird beim Auslesen gelöscht		
Zugriffsregel	Get	**Vorgabe**	N/A
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	N/A
Bereich	0 = kein Fehler 1 = ungültiger oder leerer Block 2 = Wartezeit (Wartezeit gleich) 3 = Ausführungsfehler	**Siehe auch**	Blockfehler

4.4.6 Attribut 0x06: Zähler

Dieses Attribut wird vom SERVOSTAR 300 nicht unterstützt.

Beschreibung	Globaler Zähler für Fahraufträge, kann über M LOOPCNT im Terminal abgerufen werden.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Double Integer	**Nichtflüchtig**	Nein
Bereich	Positiv	**Siehe auch**	Zähler herunterzählen (Blockobjekt)

4.5 Objekt Befehlsblock (Klasse 0x27)

Fahrsätze können über die Objektklasse “Befehlsblock“ vorab in den Antrieb programmiert werden. Diese Blocks entsprechen den SERVOSTAR-Fahraufträgen. Positionierungsbewegungen, Zeitverzögerungen und Parameteränderungen können verknüpft werden, um ein im Antrieb gespeichertes Bewegungsblockprogramm zu erstellen. Sobald das gespeicherte Blockprogramm konfiguriert wird, kann es entweder über das Objekt ”Blockfolgesteuerung" oder mit dem Blocknummernfeld Abgerufene E/A Befehlsmeldung und dem Startblockbit ausgeführt werden.

Jede Instanz der Klasse “Befehlsblock“ definiert einen speziellen Befehl. Befehlsblock Instanz #1 entspricht zum Beispiel der Bewegungsaufgabe #1 und kann vom Bildschirm Lageregler->Positionierdaten->Fahrauftragstabelle in DRIVE.EXE oder mit einem Befehl ORDER 1 vom seriellen Terminal angezeigt werden.
Die ersten beiden Attribute der Klasse “Befehlsblock” sind immer identisch: Blockbefehl und Blockverknüpfungsnummer. Beginnen Sie mit der Definition jedes Blocks durch Einstellen des Blockbefehls. Die Attribute 3-7 werden vom Wert des Blockbefehls definiert. Daher können diese erst nach der Einstellung des Blockbefehls eingestellt werden. Ein Beispiel für das Einrichten eines Bewegungsblockprogramms mit DeviceNet entnehmen Sie bitte dem Anhang.

4.5.1 Verfügbare Befehle

Der Blocktyp wird durch den Wert des ersten Attributs, Blockbefehl, definiert. Die anderen Attribute werden durch den Blockbefehl definiert; daher muss der Blockbefehl eingerichtet werden, ehe die anderen Attributwerte eingerichtet werden können.

Blockbefehl	weitere Attribute	Beschreibung
1 = Attribut ändern	Verknüpfung, Klasse, Instanz, Attribut, Daten	Legt den Wert eines für DeviceNet zugreifbaren Attributs fest.
2 = Wartezeit gleich	Verknüpfung, Klasse, Instanz, Attribut, Timeout, Daten	Verzögert, bis ein für DeviceNet zugreifbares Attribut einem gewünschten Wert entspricht.
3 = Verknüpfung größer als	Verknüpfung, Klasse, Instanz, Attribut, Alternative Verknüpfung, Daten	Testet den Wert eines Attributs und verzweigt zu einem alternativen Block, wenn der Attributwert größer als der Testwert ist.
4 = Verknüpfung weniger als	Verknüpfung, Klasse, Instanz, Attribut, Alternative Verknüpfung, Daten	Testet den Wert eines Attributs und verzweigt zu einem alternativen Block, wenn der Attributwert kleiner als der Testwert ist.
5 = Zähler herunterzählen	Verknüpfung	Dieser Block verringert den globalen Zähler im Objekt “Befehlsblockfolgesteuerung”.
6 = Verzögerung	Verknüpfung, Zeit	Dieser Block bewirkt, dass die Folgesteuerung für eine bestimmte Anzahl von Millisekunden verzögert wird, ehe mit dem nächsten Block fortgefahren wird.
8 = Trajektoriebefehl und warten	Verknüpfung, Zileposition, Zielgeschwindigkeit, Inkrementell	Führt eine Positionierungsbewegung aus.
9 = Geschwindigkeit ändern	Zielgeschwindigkeit	Führt ein Geschwindigkeitsprofil aus.

Nur der Befehl 0x08 wird vom SERVOSTAR 300 unterstützt.

4.5.2 Befehl 0x01 – Attribut ändern

Dieser Befehl wird vom SERVOSTAR 300 nicht unterstützt.

4.5.2.1 Attribut 0x01: Blockbefehl

Beschreibung	0x01 = Attribut ändern. Der Blockbefehl legt den vom Aufgabenblock durchzuführenden Befehl fest. Der Wert wird in einem anderen Format im niederwertigen Byte von O_C2 gespeichert (für weitere Informationen siehe den Anhang).		
Zugriffsregel	Get / Set	**Vorgabe**	0x01 = Befehl 01
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	N/A	**Siehe auch**	Anhang, O_C2 (ASCII), ORDER (ASCII)

4.5.2.2 Attribut 0x02: Blockverknüpfungsnummer

Beschreibung	Dieses Attribut stellt eine Verknüpfung zur nächsten, auszuführenden Blockinstanz her. Wenn dieser Block beendet ist, wird der Verbindungsblock ausgeführt. Setzen Sie dieses Attribut auf 0, um die Bewegung nach Beendigung dieser Aufgabe zu stoppen; es wird keine Folgeaufgabe ausgeführt. Der Wert wird in O_FN gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	1 bis 255	**Siehe auch**	O_FN (ASCII), ORDER (ASCII)

4.5.2.3 Attribut 0x03: Zielklasse

Beschreibung	Dieses Attribut definiert die Klassennummer des Objekts, auf das zugegriffen werden soll. Der Wert wird im höherwertigen Byte von O_ACC1 gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Datentyp	Double Integer	**Nichtflüchtig**	
Bereich		**Siehe auch**	Klasse “Lageregler”, Parameterklasse, ORDER (ASCII)

4.5.2.4 Attribut 0x04: Zielinstanz

Beschreibung	Dieses Attribut definiert die Instanznummer des Objekts, auf das zugegriffen werden soll. Der Wert wird in O_DEC1 gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Abschnitte in diesem Dokumente mit Beschreibungen der Klassenattribute, ORDER (ASCII)

4.5.2.5 Attribut 0x05: Attributnummer

Beschreibung	Dieses Attribut definiert die Attributnummer des Objekts, auf das zugegriffen werden soll. Der Wert wird im niederrwertigen Byte von O_ACC1 gespeichert. Das durch Klassen-, Instanz- und Attributnummer referenzierte Attribut im Befehl muss einstellbar sein, damit dieser Befehl ausgeführt werden kann.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Abschnitte in diesem Dokumente mit Beschreibungen der Klassenattribute, ORDER (ASCII)

4.5.2.6 Attribut 0x06: Attributdaten

Beschreibung	Dies sind die neuen Attributdaten. Der Wert wird in O_P gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	ORDER (ASCII)

4.5.3 Befehl 0x02 – Wartezeit gleich

Dieser Befehl wird vom SERVOSTAR 300 nicht unterstützt.
Mit diesem Befehl wird gewartet, bis ein Parameter einem gewünschten Wert entspricht.

4.5.3.1 Attribut 0x01: Blockbefehl

Beschreibung	0x02 = Wartezeit gleich Der Blockbefehl legt den vom Aufgabenblock durchzuführenden Befehl fest. Der Wert wird in einem anderen Format im niederwertigen Byte von O_C2 gespeichert (für weitere Informationen siehe Anhang).		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Anhang, O_C2 (ASCII), ORDER (ASCII)

4.5.3.2 Attribut 0x02: Blockverknüpfungsnummer

Beschreibung	Dieses Attribut stellt eine Verknüpfung zur nächsten, auszuführenden Blockinstanz her. Wenn dieser Block beendet ist, wird der Verbindungsblock ausgeführt. Setzen Sie dieses Attribut auf 0, um die Bewegung nach Beendigung dieser Aufgabe zu stoppen; es wird keine Folgeaufgabe ausgeführt. Der Wert wird in O_FN gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	0 bis 255	**Siehe auch**	O_FN (ASCII), ORDER (ASCII)

4.5.3.3 Attribut 0x03: Zielklasse

Beschreibung	Dieses Attribut definiert die Klassennummer des Objekts, auf das zugegriffen werden soll. Der Wert wird im höherwertigen Byte von O_ACC1 gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Klasse “Lageregler”, Parameterklasse, ORDER (ASCII)

4.5.3.4 Attribut 0x04: Zielinstanz

Beschreibung	Dieses Attribut definiert die Instanznummer des Objekts, auf das zugegriffen werden soll. Der Wert wird in O_DEC1 gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	ORDER (ASCII)

4.5.3.5 Attribut 0x05: Attributnummer

Beschreibung	Dieses Attribut definiert die Attributnummer des Objekts, auf das zugegriffen werden soll. Der Wert wird im niederrwertigen Byte von O_ACC1 gespeichert. Das durch Klassen-, Instanz- und Attributnummer referenzierte Attribut im Befehl muss einstellbar sein, damit dieser Befehl ausgeführt werden kann.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Abschnitte in diesem Dokument mit Beschreibungen der Klassenattribute, ORDER (ASCII)

4.5.3.6 Attribut 0x06: Wartezeit

Beschreibung	Maximale Wartezeit in Millisekunden, bis der Parameter dem gewünschten Wert entspricht. Ein Fehler wird gemeldet, wenn dieser Zeitgeber abgelaufen ist. Wenn er auf 0 gesetzt ist, wartet die Bewegungsaufgabe ohne Fehlermeldung. Wird in O_FT gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	O_FT (ASCII), ORDER (ASCII)

4.5.3.7 Attribut 0x07: Daten vergleichen

Beschreibung	Das Attribut wird mit diesem Wert verglichen. Wenn beide gleich sind, wird die Bewegung fortgesetzt; ansonsten wartet der Antrieb. Der Wert wird in O_P gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Abhängig von "Attributnummer"	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	ORDER (ASCII)

4.5.4 Befehl 0x03 – Bedingter Befehl "Verknüpfung größer als”

Dieser Befehl wird vom SERVOSTAR 300 nicht unterstützt.
Dieser Befehl dient für eine bedingte Verknüpfung oder Verzweigung in einer verknüpften Befehlskette. Wenn der Block ausgeführt wird, testet er den Wert eines für DeviceNet zugreifbaren Attributs und verzweigt zu einem alternativen Block, wenn der Attributwert größer als der Testwert ist.

4.5.4.1 Attribut 0x01: Blockbefehl

Beschreibung	0x03 = Größer als Test. Der Blockbefehl legt den vom Aufgabenblock durchzuführenden Befehl fest. Der Wert wird in einem anderen Format im niederwertigen Byte von O_C2 gespeichert (für weitere Informationen siehe Anhang).		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Anhang, O_C2 (ASCII), ORDER (ASCII)

4.5.4.2 Attribut 0x02: Blockverknüpfungsnummer

Beschreibung	Dieses Attribut stellt eine Verknüpfung zur nächsten, auszuführenden Blockinstanz her. Wenn dieser Block beendet ist, wird der Verbindungsblock ausgeführt. Setzen Sie dieses Attribut auf 0, um die Bewegung nach Beendigung dieser Aufgabe zu stoppen; es wird keine Folgeaufgabe ausgeführt. Der Wert wird in O_FN gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	0 bis 255	**Siehe auch**	O_FN (ASCII), ORDER (ASCII)

4.5.4.3 Attribut 0x03: Zielklasse

Beschreibung	Dieses Attribut definiert die Klassennummer des Objekts, auf das zugegriffen werden soll. Der Wert wird im höherwertigen Byte von O_ACC1 gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Klasse “Lageregler”, Parameterklasse, ORDER (ASCII)

4.5.4.4 Attribut 0x04: Zielinstanz

Beschreibung	Dieses Attribut definiert die Instanznummer des Objekts, auf das zugegriffen werden soll. Der Wert wird in O_DEC1 gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	ORDER (ASCII)

4.5.4.5 Attribut 0x05: Attributnummer

Beschreibung	Dieses Attribut definiert die Attributnummer des Objekts, auf das zugegriffen werden soll. Der Wert wird im niederrwertigen Byte von O_ACC1 gespeichert. Das durch Klassen-, Instanz- und Attributnummer referenzierte Attribut im Befehl muss einstellbar sein, damit dieser Befehl ausgeführt werden kann.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Abschnitte in diesem Dokumente mit Beschreibungen der Klassenattribute, ORDER (ASCII)

4.5.4.6 Attribut 0x06: Verknüpfungsnummer vergleichen

Beschreibung	Wenn der Attributwert größer als der Testwert ist, verzweigen Sie zu dem in diesem Attribut angegebenen Attribut anstatt zu dem in Attribut 2 angegebenen Block. Der Wert wird in O_DEC2 gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	1 bis 255	**Siehe auch**	ORDER (ASCII)

4.5.4.7 Attribut 0x07: Daten vergleichen

Beschreibung	Dieses Attribut vergleicht die Daten für die bedingte Verknüpfung. Wenn das Testattribut größer ist als die Vergleichsdaten, wird die normale Verknüpfung (Attribut 2) ignoriert und der nächste, ausgeführte Block ist der Vergleichsverknüpfungsblock (Attribut 6).		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Abhängig von "Attributnummer"	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	1 bis 255	**Siehe auch**	ORDER (ASCII)

4.5.5 Befehl 0x04 – Befehl "Bedingte Verknüpfung weniger als"

Dieser Befehl wird vom SERVOSTAR 300 nicht unterstützt.
Dieser Befehl dient für eine bedingte Verknüpfung oder Verzweigung in einer verknüpften Befehlskette. Wenn der Block ausgeführt wird, testet er den Wert eines für DeviceNet zugreifbaren Attributs und verzweigt zu einem alternativen Block, wenn der Attributwert kleiner als der Testwert ist.

4.5.5.1 Attribut 0x01: Blockbefehl

Beschreibung	0x04 = Kleiner als Test. Der Blockbefehl legt den vom Aufgabenblock durchzuführenden Befehl fest. Der Wert wird in einem anderen Format im niederwertigen Byte von O_C2 gespeichert (für weitere Informationen siehe Anhang).		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Anhang, O_C2 (ASCII), ORDER (ASCII)

4.5.5.2 Attribut 0x02: Blockverknüpfungsnummer

Beschreibung	Dieses Attribut stellt eine Verknüpfung zur nächsten, auszuführenden Blockinstanz her. Wenn dieser Block beendet ist, wird der Verbindungsblock ausgeführt. Dieser Wert darf für den Verzögerungsbefehl nicht Null sein. Der Wert wird in O_FN gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	1 bis 255	**Siehe auch**	O_FN (ASCII), ORDER (ASCII)

4.5.5.3 Attribut 0x03: Zielklasse

Beschreibung	Dieses Attribut definiert die Klassennummer des Objekts, auf das zugegriffen werden soll. Der Wert wird im höherwertigen Byte von O_ACC1 gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Klasse “Lageregler”, Parameterklasse, ORDER (ASCII)

4.5.5.4 Attribut 0x04: Zielinstanz

Beschreibung	Dieses Attribut definiert die Instanznummer des Objekts, auf das zugegriffen werden soll. Der Wert wird in O_DEC1 gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	ORDER (ASCII)

4.5.5.5 Attribut 0x05: Attributnummer

Beschreibung	Dieses Attribut definiert die Attributnummer des Objekts, auf das zugegriffen werden soll. Der Wert wird im niederrwertigen Byte von O_ACC1 gespeichert. Das durch Klassen-, Instanz- und Attributnummer referenzierte Attribut im Befehl muss einstellbar sein, damit dieser Befehl ausgeführt werden kann.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Abschnitte in diesem Dokumente mit Beschreibungen der Klassenattribute, ORDER (ASCII)

4.5.5.6 Attribut 0x06: Verknüpfungsnummer vergleichen

Beschreibung	Wenn der Attributwert kleiner als der Testwert ist, verzweigen Sie zu dem in diesem Attribut angegebenen Attribut anstatt zu dem in Attribut 2 angegebenen Block. Der Wert wird in O_DEC2 gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	1 bis 255	**Siehe auch**	ORDER (ASCII)

4.5.5.7 Attribut 0x07: Daten vergleichen

Beschreibung	Dieses Attribut vergleicht die Daten für die bedingte Verknüpfung. Wenn das Testattribut unter den Vergleichsdaten liegt, wird die normale Verknüpfung (Attribut 2) ignoriert und der nächste, ausgeführte Block ist der Vergleichsverknüpfungsblock (Attribut 6).		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Abhängig von "Attributnummer"	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	ORDER (ASCII)

4.5.6 Befehl 0x05 – Zähler herunterzählen

Dieser Befehl wird vom SERVOSTAR 300 nicht unterstützt.
Dieser Befehl dient zum Herunterzählen des globalen Zählers (Klasse “Blockfolgesteuerung” 0x26, Instanz 1, Attribut 6). Kombinieren Sie diesen Block mit “Attribut ändern” und “Kleiner als Test”-Blocks, um Schleifen und Verzweigungen innerhalb Ihres Blockprogramms zu implementieren.

4.5.6.1 Attribut 0x01: Blockbefehl

Beschreibung	0x05 = Zähler herunterzählen Der Blockbefehl legt den vom Aufgabenblock durchzuführenden Befehl fest. Der Wert wird in einem anderen Format im niederwertigen Byte von O_C2 gespeichert (für weitere Informationen siehe Anhang).		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Anhang, O_C2 (ASCII), ORDER (ASCII)

4.5.6.2 Attribut 0x02: Blockverknüpfungsnummer

Beschreibung	Dieses Attribut stellt eine Verknüpfung zur nächsten, auszuführenden Blockinstanz her. Wenn dieser Block beendet ist, wird der Verbindungsblock ausgeführt. Setzen Sie dieses Attribut auf 0, um die Bewegung nach Beendigung dieser Aufgabe zu stoppen; es wird keine Folgeaufgabe ausgeführt. Der Wert wird in O_FN gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	0 bis 255	**Siehe auch**	O_FN (ASCII), ORDER (ASCII)

4.5.7 Befehl 0x06 – "Verzögerung"

Dieser Befehl wird vom SERVOSTAR 300 nicht unterstützt.
Mit diesem Befehl wird eine verknüpfte Befehlskette verzögert.

4.5.7.1 Attribut 0x01: Blockbefehl

Beschreibung	0x06 = Verzögerung. Der Blockbefehl legt den vom Aufgabenblock durchzuführenden Befehl fest. Der Wert wird im niederwertigen Byte von O_C2 gespeichert (für weitere Informationen siehe Anhang).		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Anhang, O_C2 (ASCII), ORDER (ASCII)

4.5.7.2 Attribut 0x02: Blockverknüpfungsnummer

Beschreibung	Dieses Attribut stellt eine Verknüpfung zur nächsten, auszuführenden Blockinstanz her. Wenn dieser Block beendet ist, wird der Verbindungsblock ausgeführt. Setzen Sie dieses Attribut auf 0, um die Bewegung nach Beendigung dieser Aufgabe zu stoppen; es wird keine Folgeaufgabe ausgeführt. Der Wert wird in O_FN gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	0 bis 255	**Siehe auch**	O_FN (ASCII), ORDER (ASCII)

4.5.7.3 Attribut 0x03: Verzögerung

Beschreibung	Dieses Attribut legt die Verzögerung in Millisekunden fest. Der Wert wird in O_FT gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	O_FT (ASCII), ORDER (ASCII)

4.5.8 Befehl 0x08 – Fahrauftrag

Mit diesem Befehl wird eine Positionierungsbewegung ausgelöst und auf deren Abschluss gewartet. Die Beschleunigung und die Verzögerung werden in O_ACC1 und O_DEC1 von ORDER 0 gespeichert. Bits 0x800 in O_C und 0x100 in O_C2 der Aufgabe werden auf 1 gesetzt, so dass die Beschleunigung und die Verzögerung von Aufgabe 0 und nicht von der aktuellen Aufgabe genommen werden. Dies ermöglicht globale Werte für DeviceNet-Bewegungsblocks.

4.5.8.1 Attribut 0x01: Blockbefehl

Beschreibung	0x08 = Trajektorie initiieren. Der Blockbefehl legt den vom Aufgabenblock durchzuführenden Befehl fest. Der Wert wird in einem anderen Format im niederwertigen Byte von O_C2 gespeichert (für weitere Informationen siehe Anhang).		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Anhang, O_C2 (ASCII), ORDER (ASCII)

4.5.8.2 Attribut 0x02: Blockverknüpfungsnummer

Beschreibung	Dieses Attribut stellt eine Verknüpfung zur nächsten, auszuführenden Blockinstanz her. Wenn dieser Block beendet ist, wird der Verbindungsblock ausgeführt. Setzen Sie dieses Attribut auf 0, um die Bewegung nach Beendigung dieser Aufgabe zu stoppen; es wird keine Folgeaufgabe ausgeführt. Der Wert wird in O_FN gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	0 bis 255	**Siehe auch**	O_FT (ASCII), ORDER (ASCII)

4.5.8.3 Attribut 0x03: Zielposition

Beschreibung	Definiert die Zielprofilposition in Positionseinheiten. Der Wert wird in O_P gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	O_P (ASCII), ORDER (ASCII)

4.5.8.4 Attribut 0x04: Zielgeschwindigkeit

Beschreibung	Dieses Attribut definiert die Zielprofilpositionsgeschwindigkeit in Profileinheiten pro Sekunde. Der Wert wird in O_V gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	O_V (ASCII), ORDER (ASCII)

4.5.8.5 Attribut 0x05: Inkrementell

Beschreibung	Dieses Flag legt fest, ob die Bewegung inkrementell oder absolut ist.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Boolean	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	0 = absolute Position 1 = inkrementelle Position	**Siehe auch**	O_C (ASCII), ORDER (ASCII)

4.5.8.6 Attribut 0x64: O_C

Dieses Attribut wird vom SERVOSTAR 600 nicht unterstützt.

Beschreibung	Dieses Attribut ermöglicht direkten Zugriff auf den Parameter O_C ORDER. O_C wird auch automatisch geändert, wenn der Blockbefehl (Objekt Befehlsblock, Attribut 0x01) auf Befehl 0x08 gesetzt wird.		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Long Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	O_C (ASCII), ORDER (ASCII)

4.5.8.7 Attribut 0x65: O_ACC

Dieses Attribut wird vom SERVOSTAR 600 nicht unterstützt.

Beschreibung	Dieses Attribut ermöglicht direkten Zugriff auf den Parameter O_ACC ORDER. O_ACC wird auch automatisch geändert, wenn der Blockbefehl (Objekt Befehlsblock, Attribut 0x01) auf Befehl 0x08 gesetzt wird.		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Long Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	O_ACC (ASCII), ORDER (ASCII)

4.5.8.8 Attribut 0x66: O_DEC

Dieses Attribut wird vom SERVOSTAR 600 nicht unterstützt.

Beschreibung	Dieses Attribut ermöglicht direkten Zugriff auf den Parameter O_DEC ORDER. O_DEC wird auch automatisch geändert, wenn der Blockbefehl (Objekt Befehlsblock, Attribut 0x01) auf Befehl 0x08 gesetzt wird.		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Long Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	O_DEC (ASCII), ORDER (ASCII)

4.5.8.9 Attribut 0x67: O_TAB

Dieses Attribut wird vom SERVOSTAR 600 nicht unterstützt.

Beschreibung	Dieses Attribut ermöglicht direkten Zugriff auf den Parameter O_TAB ORDER. O_TAB wird auch automatisch geändert, wenn der Blockbefehl (Objekt Befehlsblock, Attribut 0x01) auf Befehl 0x08 gesetzt wird.		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Long Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	O_TAB (ASCII), ORDER (ASCII)

4.5.8.10 Attribut 0x68: O_FT

Dieses Attribut wird vom SERVOSTAR 600 nicht unterstützt.

Beschreibung	Dieses Attribut ermöglicht direkten Zugriff auf den Parameter O_FT ORDER. O_FT wird auch automatisch geändert, wenn der Blockbefehl (Objekt Befehlsblock, Attribut 0x01) auf Befehl 0x08 gesetzt wird.		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Long Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	O_FT (ASCII), ORDER (ASCII)

4.5.9 Befehl 0x09 – Tippbetrieb

Dieser Befehl wird vom SERVOSTAR 300 nicht unterstützt.
Dieser Befehl dient zum Ausführen eines Geschwindigkeitsprofils. Da die Bewegung von unbegrenzter Dauer ist (und andauert, bis sie gestoppt wird), kann der Block nicht mit einer Folgeaufgabe verknüpft werden.

4.5.9.1 Attribut 0x01: Blockbefehl

Beschreibung	0x09 = Geschwindigkeitsänderung. Der Blockbefehl legt den vom Aufgabenblock durchzuführenden Befehl fest. Der Wert wird in einem anderen Format im niederwertigen Byte von O_C2 gespeichert (für weitere Informationen siehe Anhang).		
Zugriffsregel	Get / Set	**Vorgabe**	N/A
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	Anhang, O_C2 (ASCII), ORDER (ASCII)

4.5.9.2 Attribut 0x02: Blockverknüpfungsnummer

Beschreibung	Dieses Attribut stellt normalerweise eine Verknüpfung zur nächsten, auszuführenden Blockinstanz her. Da ein Befehl zur Geschwindigkeitsänderung keinen Folgeblock aufweisen kann, sollte dieses Attribut nicht gesetzt werden.		
Zugriffsregel	Get / Set	**Vorgabe**	0
Datentyp	Unsigned Short Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich	0	**Siehe auch**	O_FT (ASCII), ORDER (ASCII)

4.5.9.3 Attribut 0x03: Zielgeschwindigkeit

Beschreibung	Dieses Attribut definiert die Zielprofilpositionsgeschwindigkeit in Profileinheiten pro Sekunde. Der Wert wird in O_V gespeichert.		
Zugriffsregel	Get / Set	**Vorgabe**	
Datentyp	Double Integer	**Nichtflüchtig**	Instanzen 1-180 werden nichtflüchtig, Instanzen 181-255 flüchtig gespeichert
Bereich		**Siehe auch**	O_V (ASCII), ORDER (ASCII)

4.6 Objekt "Digitaler Eingang" (Klasse 0x08)

Dieses Objekt ermöglicht den Zugriff auf die vier digitalen Eingänge des Antriebs. Die Instanzen 1-4 entsprechen den digitalen Eingängen 1-4.

4.6.1 Attribut 0x03: Wert

Beschreibung	Dieses Attribut lautet 1, wenn der digitale Eingang hoch ist. Die Instanzen 1-4 entsprechen den digitalen Eingängen 1-4.		
Zugriffsregel	Get	**Vorgabe**	keine
Datentyp	Boolean	**Nichtflüchtig**	N/A
Bereich	0 = Eingang ist inaktiv 1 = Eingang ist aktiv	**Siehe auch**	IN1 - IN4 (ASCII)

4.7 Objekt "Digitaler Ausgang" (Klasse 0x09)

Dieses Objekt ermöglicht den Zugriff auf die beiden digitalen Ausgänge des Antriebs. Die Instanzen 1-2 entsprechen den digitalen Ausgängen 1-2. Zum Konfigurieren des Antriebs für die DeviceNet-Steuerung der digitalen Ausgänge setzen Sie O1MODE=23 und O2MODE=23.

4.7.1 Attribut 0x03: Wert

Beschreibung	Setzt Sie dieses Attribut auf 1, um den digitalen Ausgang auf Hoch zu setzen. Die Instanzen 1-2 entsprechen digitalen Ausgängen.		
Zugriffsregel	Set	**Vorgabe**	keine
Datentyp	Boolean	**Nichtflüchtig**	N/A
Bereich	0 = Ausgang auf low setzen 1 = Ausgang auf high setzen	**Siehe auch**	O1MODE / O2MODE (ASCII), O1 / O2 (ASCII)

4.8 Objekt "Analoger Eingang" (Klasse 0x0A)

Dieses Objekt ermöglicht den Zugriff auf die beiden analogen Eingänge des Antriebs. Die Instanzen 1-2 entsprechen den analogen Eingängen 1-2.

4.8.1 Attribut 0x03: Wert

Beschreibung	Spannung am analogen Eingang in Millivolt. Die Instanzen 1-2 entsprechen den analogen Eingängen 1-2.		
Zugriffsregel	Get	**Vorgabe**	keine
Datentyp	Integer	**Nichtflüchtig**	N/A
Bereich	-10000 bis 10000	**Siehe auch**	ANIN1 / ANIN2 (ASCII)

4.9 Objekt "Analoger Ausgang" (Klasse 0x0B)

Dieses Objekt wird vom SERVOSTAR 300 nicht unterstützt
Dieses Objekt ermöglicht den Zugriff auf die beiden analogen Ausgänge des Antriebs. Die Instanzen 1-2 entsprechen den analogen Ausgängen 1-2. Zum Konfigurieren des Antriebs für die DeviceNet-Steuerung der analogen Ausgänge setzen Sie ANOUT1=6 und ANOUT2=6.

4.9.1 Attribut 0x03: Wert

Beschreibung	Einstellen auf die gewünschte Ausgangsspannung in Millivolt. Zum Konfigurieren des Antriebs für die DeviceNet-Steuerung der analogen Ausgänge setzen Sie ANOUT1=6 und ANOUT2=6. Der Wert wird in AN1TRIG / AN2TRIG gespeichert.		
Zugriffsregel	Set	**Vorgabe**	keine
Datentyp	Integer	**Nichtflüchtig**	N/A
Bereich	-1000 bis 10000	**Siehe auch**	ANOUT1 / ANOUT2 (ASCII), AN1TRIG / AN2TRIG (ASCII)

4.10 Objekt "Identität" (Klasse 0x01)

Objekt "Identität" 0x01					
Objektklasse	**ID**	**Beschreibung**	**Get**	**Set**	**Grenzwerte**
Attribute Offen	1	Revision			
	2	Max. Instanz			
X Keine Unterstützung	3	Anzahl Instanzen			
	4	Optionale Attributliste			
	5	Optionale Serviceliste			
	6	Max. ID der Klassenattribute			
	7	Max. ID der Instanzattribute			
	DeviceNet Services		**Parameteroptionen**		
Services		Get_Attributes_All			
		Reset			
X Keine Unterstützung		Get_Attribute_Single			
		Find_Next_Object_instance			
Objektinstanz	**ID**	**Beschreibung**	**Get**	**Set**	**Grenzwerte**
Attribute Offen	1	Lieferant	X		=(452)
	2	Gerätetyp	X		=(16)
	3	Produktcode	X		=(3)
	4	Revision	X		=(1.1)
	5	Status (unterstützte Bit)	X		
	6	Seriennummer	X		
	7	Produktname	X		SERVOSTAR 603
	8	Status			
	9	Konfigurationskonsistenzwert			
	10	Heartbeat-Intervall			
	DeviceNet Services		**Parameteroptionen**		
Services		Get_Attributes_All			
	X	Reset	0,1		
	X	Get_Attribute_Single			
		Set_Attribute_Single			

4.11 Objekt "Message Router" (Klasse 0x02)

Objekt "Message Router" 0x02					
Objektklasse	**ID**	**Beschreibung**	**Get**	**Set**	**Grenzwerte**
Attribute Offen	1	Revision			
	2	Max. Instanz			
X Keine Unterstützung	3	Anzahl Instanzen			
	4	Optionale Attributliste			
	5	Optionale Serviceliste			
	6	Max. ID der Klassenattribute			
	7	Max. ID der Instanzattribute			
	DeviceNet Services		**Parameteroptionen**		
Services		Get_Attributes_All			
X Keine Unterstützung		Get_Attribute_Single			
Objektinstanz	**ID**	**Beschreibung**	**Get**	**Set**	**Grenzwerte**
Attribute Offen	1	Objektliste			
	2	Maximal unterstützte Verbindungen			
X Keine Unterstützung	3	Anzahl aktiver Verbindungen			
	4	Liste der aktiven Verbindungen			
	DeviceNet Services		**Parameteroptionen**		
Services		Get_Attributes_All			
		Get_Attribute_Single			
X Keine Unterstützung		Set_Attribute_Single			

4.12 Objekt "DeviceNet" (Klasse 0x03)

Objekt "DeviceNet" 0x03					
Objektklasse	**ID**	**Beschreibung**	**Get**	**Set**	**Grenzwerte**
Attribute Offen	1	Revision	X		
	2	Max. Instanz			
Keine Unterstützung	3	Anzahl Instanzen			
	4	Optionale Attributliste			
	5	Optionale Serviceliste			
	6	Max. ID der Klassenattribute			
	7	Max. ID der Instanzattribute			
	DeviceNet Services		**Parameteroptionen**		
Services	X	Get_Attribute_Single			
Keine Unterstützung					
Objektinstanz	**ID**	**Beschreibung**	**Get**	**Set**	**Grenzwerte**
Attribute Offen	1	MAC ID	X		
	2	Baudrate	X		
Keine Unterstützung	3	BOI	X	X	
	4	Zähler "Kommunikationsabbruch"	X	X	
	5	Zuordnungsinformationen	X		
	6	Schalter MAC ID geändert			
	7	Schalter für Baudrate geändert			
	8	Schalterwert MAC ID	X		
	9	Schalterwert Baudrate	X		
	DeviceNet Services		**Parameteroptionen**		
Services	X	Get_Attributes_All			
	X	Set_Attribute_Single			
Keine Unterstützung	X	"M/S Verbindung zuweisen" gesetzt			
	X	"M/S Verbindung freigeben" gesetzt			

4.13 Objekt "Verbindung" (Klasse 0x0 5) - Explizit

<table>
<tr><th colspan="10">Objekt "Verbindung" 0x05</th></tr>
<tr><th>Objektklasse</th><th>ID</th><th colspan="2">Beschreibung</th><th colspan="2">Get</th><th colspan="2">Set</th><th colspan="2">Grenzwerte</th></tr>
<tr><td>Attribute Offen</td><td>1</td><td colspan="2">Revision</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>2</td><td colspan="2">Max. Instanz</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>X Keine Unterstützung</td><td>3</td><td colspan="2">Anzahl Instanzen</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>4</td><td colspan="2">Optionale Attributliste</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>5</td><td colspan="2">Optionale Serviceliste</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>6</td><td colspan="2">Max. ID der Klassenattribute</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>7</td><td colspan="2">Max. ID der Instanzattribute</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><th></th><th colspan="3">DeviceNet Services</th><th colspan="6">Parameteroptionen</th></tr>
<tr><td>Services</td><td></td><td colspan="8">Reset</td></tr>
<tr><td></td><td></td><td colspan="8">Create</td></tr>
<tr><td></td><td></td><td colspan="8">Delete</td></tr>
<tr><td>X Keine Unterstützung</td><td></td><td colspan="8">Get_Attribute_Single</td></tr>
<tr><td></td><td></td><td colspan="8">Find_Next_Object_instance</td></tr>
<tr><th>Objektinstanz</th><th colspan="2">Anschlusstyp</th><th colspan="7">Max. Verbindungsinstanzen</th></tr>
<tr><td></td><td colspan="2">M/S explizite Meldung</td><td colspan="2">1 Server</td><td colspan="3">Client</td><td colspan="2">1 gesamt</td></tr>
<tr><td></td><td colspan="2">Produktionsauslösung(en)</td><td>Zykl.</td><td></td><td colspan="2">COS</td><td></td><td>Anwen-
dungsausl.</td><td></td></tr>
<tr><td></td><td colspan="2">Transporttyp(en)</td><td>Server</td><td>X</td><td colspan="2"></td><td></td><td>Client</td><td></td></tr>
<tr><td></td><td colspan="2">Transportklasse(n)</td><td></td><td></td><td colspan="2">2</td><td></td><td>3</td><td>X</td></tr>
<tr><th></th><th>ID</th><th colspan="2">Beschreibung</th><th colspan="2">Get</th><th colspan="2">Set</th><th colspan="2">Grenzwerte</th></tr>
<tr><td>Attribute Offen</td><td>1</td><td colspan="2">Status</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>2</td><td colspan="2">Instanztyp</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>3</td><td colspan="2">Transportklassenauslösung</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>4</td><td colspan="2">ID der hergestellten Verbindung</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>5</td><td colspan="2">ID der verwendeten Verbindung</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>6</td><td colspan="2">Ursprüngliche Komm.eigenschaften</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>7</td><td colspan="2">Größe der hergestellten Verbindung</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>8</td><td colspan="2">Größe der verwendeten Verbindung</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>9</td><td colspan="2">Erwartete Paketgeschwindigkeit</td><td colspan="2">X</td><td colspan="2">X</td><td colspan="2"></td></tr>
<tr><td></td><td>12</td><td colspan="2">Wartezeit-Aktion Watchdog</td><td colspan="2">X</td><td colspan="2">X</td><td colspan="2"></td></tr>
<tr><td></td><td>13</td><td colspan="2">Pfadlänge der hergestellten Verb.</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>14</td><td colspan="2">Pfad der hergestellten Verbindung</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>15</td><td colspan="2">Pfadlänge der verwendeten Verb.</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>16</td><td colspan="2">Pfad der verwendeten Verbindung</td><td colspan="2">X</td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td></td><td>17</td><td colspan="2">Produktionssperrzeit</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><th></th><th colspan="3">DeviceNet Services</th><th colspan="6">Parameteroptionen</th></tr>
<tr><td>Services</td><td>X</td><td colspan="8">Reset</td></tr>
<tr><td></td><td></td><td colspan="8">Delete</td></tr>
<tr><td></td><td></td><td colspan="8">Apply_attributes</td></tr>
<tr><td></td><td>X</td><td colspan="8">Get_Attribute_Single</td></tr>
<tr><td></td><td>X</td><td colspan="8">Set_Attribute_Single</td></tr>
</table>

4.14 Objekt "Verbindung" (Klasse 0x05) - abgefragter E/A

Objekt "Verbindung" 0x05

Objektklasse	ID	Beschreibung	Get	Set	Grenzwerte
Attribute Offen	1	Revision			
	2	Max. Instanz			
X Keine Unterstützung	3	Anzahl Instanzen			
	4	Optionale Attributliste			
	5	Optionale Serviceliste			
	6	Max. ID der Klassenattribute			
	7	Max. ID der Instanzattribute			
	DeviceNet Services		**Parameteroptionen**		
Services	X	Reset			
		Create			
		Delete			
X Keine Unterstützung		Get_Attribute_Single			
		Find_Next_Object_instance			

Objektinstanz	Anschlusstyp	Max. Verbindungsinstanzen					
	M/S Abfrage	1 Server		Client		1 gesamt	
	Produktionsauslösung(en)	Zykl.	X	COS		Anwendungstrigger	
	Transporttyp(en)	Server	X			Client	
	Transportklasse(n)			2	X	3	

	ID	Beschreibung	Get	Set	Grenzwerte
Attribute Offen	1	Status	X		
	2	Instanztyp	X		
	3	Transportklassenauslösung	X		
	4	ID hergestellte Verbindung	X		
	5	ID der verwendeten Verbindung	X		
	6	Ursprüngliche Komm.eigenschaften	X		
	7	Größe der hergestellten Verbindung	X		
	8	Größe der verwendeten Verbindung	X		
	9	Erwartete Paketgeschwindigkeit	X	X	
	12	Wartezeit-Aktion Watchdog	X	X	
	13	Pfadlänge der hergestellten Verb.	X		
	14	Pfad der hergestellten Verbindung	X		
	15	Pfadlänge der verwendeten Verb.	X		
	16	Pfad der verwendeten Verbindung	X		
	17	Produktionssperrzeit			
	DeviceNet Services		**Parameteroptionen**		
Services	X	Reset			
		Delete			
		Apply_attributes			
	X	Get_Attribute_Single			
	X	Set_Attribute_Single			

5 Gruppenmeldungen

Normalerweise werden Gruppenmeldungen (Abgerufene E/A) für Echtzeitdaten und Bewegungssteuerung verwendet. Für weitere Informationen siehe Abschnitt.3.2

5.1 Abgerufene E/A-Befehlsgruppen

Abgerufene E/A Messaging ist eine Methode zum Übertragen einer Gruppe von Steuer-Bits und eines Datenbefehls und Erhalten einer Gruppe von Status-Bits und einer Antwort mit einem Datenwert. Diese Kommunikationsmethode ist bevorzugt, da mit Explicit Messaging nur jeweils ein Wert übertragen werden kann. Abgerufene E/A und Explicit Messaging können gleichzeitig für die Kommunikation zwischen dem Regler und dem Antrieb verwendet werden. In diesem Abschnitt wird das Format für jede Befehlsgruppe definiert. Außerdem enthält dieser Abschnitt Beispiele für jede Befehlsgruppe.

Befehlsgruppen enthalten Steuer-Bits, die für jeden Befehlstyp identisch definiert sind. Zusätzlich zu den Steuer-Bits kann eine Befehlsgruppe zum Senden von jeweils einem Datenbefehl verwendet werden (Zielposition, Zielgeschwindigkeit, Beschleunigung, Geschwindigkeitsabnahme oder Drehmoment). Der Befehlstyp wird im Feld Befehlsgruppentyp angegeben.

Der Antrieb reagiert auf jede eingegangene Befehlsgruppe mit der Übertragung einer Antwortgruppe (eine Beschreibung dazu finden Sie im folgenden Abschnitt 5.2. Abgerufene E/A Antwortgruppen). Die Antwortgruppe enthält Steuer-Bits, die für jeden Antworttyp identisch definiert sind. Zusätzlich zu den Status-Bits kann eine Antwortgruppe jeweils einen Datenwert übertragen (Ist-Position, befohlene Position, Ist-Geschwindigkeit, Ist-Drehmoment oder Fehlercode). Der Antworttyp wird im Feld Antwortgruppentyp der Befehlsgruppe angegeben. Eine Befehlsgruppe kann sowohl einen Befehlsgruppentyp und einen Antwortgruppentyp enthalten, um einen Befehl zu übertragen und eine Antwort in der selben Gruppe anzufordern.



Alle acht Datenbyte werden ignoriert, wenn kein gültiger Gruppentyp in Byte 2 angegeben ist. (Gültige Befehlsgruppentypen sind 0 bis 5.)

Daten außerhalb des Bereichs des Attributs bewirken eine Fehlerantwortgruppe. Dies gilt für alle Befehlsgruppen außer Gruppe 5 (Drehmoment).

Der Antrieb muss referenziert werden, bevor eine Bewegung beginnt. Geschieht dies nicht, löst der Antrieb einen Antriebsalarm aus. Dieser Alarm muss gelöscht werden, bevor der Antrieb wieder in Betrieb genommen werden kann.

5.1.1 Steuer-Bits und Datenfelder

<table>
<tr><th>Byte</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td>0</td><td>Aktivieren</td><td>Reg. aktivieren</td><td>Sofortiger Stopp</td><td>Kontrollierter Stopp</td><td>Richtung</td><td>Inkrementell</td><td>Startblock</td><td>Laden / Start</td></tr>
<tr><td>1</td><td colspan="8">Blocknummer</td></tr>
<tr><td>2</td><td colspan="3">Befehlsachse = 001</td><td colspan="5">Typ der Eingangsbefehlsgruppe</td></tr>
<tr><td>3</td><td colspan="3">Antwortachse = 001</td><td colspan="5">Typ der Ausgangsantwortgruppe</td></tr>
<tr><td>4</td><td colspan="8">Daten niederwertiges Byte</td></tr>
<tr><td>5</td><td colspan="8">Daten niederwertiges, mittleres Byte</td></tr>
<tr><td>6</td><td colspan="8">Daten höherwertiges, mittleres Byte</td></tr>
<tr><td>7</td><td colspan="8">Daten höherwertiges Byte</td></tr>
</table>

Aktivieren Das Setzen dieses Bit aktiviert den Verstärker. Siehe auch Enable (Klasse 0x25 Lageregler, Attribut 0x11).

Registrierung aktivieren Registrierungseingang aktivieren.
Dieses Bit wird vom SERVOSTAR 300 nicht unterstützt

Sofortiger Stopp Wenn dieses Bit gesetzt ist, wird der Antrieb sofort (ohne Verzögerung) gestoppt.
Siehe auch “Sofortiger Stopp” (Klasse 0x25 Lageregler, Attribut 0x15).

Kontrollierter Stopp Wenn dieses Bit gesetzt ist, wird der Antrieb langsam gestoppt. Siehe auch “Kontrollierter Stopp” (Klasse 0x25 Lageregler, Attribut 0x14).



Geben Sie nur HARD STOP (sofortiger Stopp) oder nur SMOOTH STOP (kontrollierter Stopp) aus, um die Bewegung zu stoppen. Wenn Sie eines dieser Bits zur gleichen Zeit wie das Bit "Trajektoriestart" ändern, führt dies zu unvorhergesehenen Aktionen der Steuerung.

Richtung Dieses Bit wird nur im Geschwindigkeitsmodus verwendet. Positiv = 1, Negativ = 0 Siehe auch Objekt "Richtung" (ageregler, Klasse 0x25, Attribut 0x17).

Inkrementell Dieses Bit wird nur im Positionsmodus verwendet. Dieses Bit zeigt an, ob die in Byte 4 bis 7 der Befehlsgruppe 1 "Zielposition" festgelegte Position absolut (0) oder relativ (1) ist. Siehe Beschreibung des Flag für "Relativmode".

Startblock Wenn dieses Bit hoch (1) und die Blocknummer auf Null (0) gesetzt ist, werden zuvor erzeugte Programme ausgeführt und im Antrieb gespeichert. Das ausgeführte Programm wird durch die letzten vier Byte der Befehlsgruppe definiert. Um die Programmausführung zu stoppen, müssen Sie den Startblock hoch (1) und die Blocknummer hoch (1) setzen.
Programme können mit jeder Befehlsgruppe ausgeführt werden. Siehe beigefügtes Beispiel.



Eine unkontrollierte Aktion tritt ein, wenn der Startblock hoch (1) gesetzt und gleichzeitig ein Zustandswechsel von 0 auf 1 für "Trajektoriestart" ausgegeben wird.

Laden/Start Durch den Zustandswechsel dieses Bit von 0 auf 1 wird eine Bewegung in Befehlsgruppe 1 "Zielposition" gestartet. Für alle anderen Befehlsgruppen legt der Wechsel dieses Bit den Datenwert fest (d.h. Geschwindigkeit, Beschleunigung usw.). Siehe auch Objektklasse "Trajektoriestart, Lageregler” (ID=37).

Blocknummer Dient, zusammen mit “Startblock“ zur Ausführung einer zuvor im Antrieb definierten Blockbefehlsfolge (Fahrsatz). Dieses Feld gibt an, dass die Blockinstanz mit der Ausführung beginnen soll, wenn der Startblock von 0 auf 1 wechselt. Das Feld Blocknummer wird nur verwendet um Blockbefehle auszuführen, nicht um sie zu verändern. Um Blockbefehle zu editieren, senden Sie explizite Objekte an den Blockbefehl. Siehe auch ”Block" (Klasse 0x26 “Blockfolgesteuerung”, Attribut 0x01).

Befehlsachse Der Antrieb unterstützt nur eine Achse; daher muss dieser Wert immer 1 sein. Alle anderen Werte verursachen die Fehlerantwort COMMAND_AXIS_INVALID.

Antwortachse Der Antrieb unterstützt nur eine Achse; daher muss dieser Wert immer 1 sein. Alle anderen Werte verursachen die Fehlerantwort RESPONSE_AXIS_INVALID.

Befehlsgruppen-typ Die Zielposition, Geschwindigkeit, Beschleunigung, Verzögerung und Drehmoment können mit einer Befehlsgruppe modifiziert werden. Setzen Sie den Befehlstyp auf die gewünschte Befehlszahl (wie in den folgenden Abschnitten beschrieben). Setzen Sie den Befehlstyp auf Null (0), um keinen Befehl in der Gruppe zu geben.

Antwortgruppen-typ Setzen Sie den Antworttyp in der Befehlsgruppe, um festzulegen, welche Daten in der Antwortgruppe zurückgeliefert werden. Ist-Position, Zielposition, ist-Geschwindigkeit und Ist-Drehmoment sind verfügbar. Siehe Abgerufene E/A Antwortgruppe für weitere Informationen. Setzen Sie den Antworttyp auf Null (0), um keine Datenantwort anzufordern (eine Antwortgruppe mit gültigen Status-Bits wird dennoch zurückgeliefert).

Daten-Bytes Laden Sie Daten für den gewünschten Befehlstyp in die Datenfelder, mit dem niederwertigsten Byte zuerst.

5.1.2 Ausführen einer gespeicherten Sequenz über DeviceNet

Eine Folge von Fahrsätzen kann im Programm des Antriebs (Graphical Motion Tasking) oder über DeviceNet konfiguriert (siehe Objekt "Befehlsblock") und später über DeviceNet ausgeführt werden. Weitere Anweisungen zum Erstellen einer Folge von Fahrsätzen finden Sie im Handbuch zur Inbetriebnahmesoftware und der Onlinehilfe.

Zur Durchführung einer Fahrsatzfolge setzen Sie die Blocknummer gleich dem Index des zur Ausführung anstehenden Blocks. Das Bit "Startblock" muss eine positive Flanke aufweisen. "Aktivieren" muss ebenfalls eine positive Flanke aufweisen, während die Stoppbits eine negative Flanke haben müssen.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Aktivieren	Reg. aktivieren	Sofortiger Stopp	Kontrollierter Stopp	Richtung (Geschwindig-keitsmodus)	Inkrementell	Startblock	Trajekto-riestart
1	Blocknummer							
2	0	0	1	0	Typ der Eingangsbefehlsgruppe (0000)			
3	0	0	1	0	Typ der Ausgangsantwortgruppe			
4	0							
5	0							
6	0							
7	0							

Zum Stoppen der Ausführung einer Befehlssequenz setzen Sie das Bit für "kontrollierten Stopp" oder "sofortigen Stopp" hoch.

5.1.3 Data Handshaking

Data Handshaking dient zum Übertragen von Datenbefehlen mit Abgerufene E/A Messaging. Zum Übertragen eines Befehls an den Antrieb setzen Sie den Befehlstyp und laden Sie Daten in die Datenfelder und wechseln Sie dann das Lade-/Start-Bit auf Hoch. Der Antrieb akzeptiert Daten nur, wenn Laden/Starten von 0 auf 1 wechselt. Wenn die Daten erfolgreich geladen wurden, setzt der Antrieb das Antwortflag “Laden abgeschlossen” auf Hoch. “Laden abgeschlossen” wird vom Antrieb gelöscht, nachdem Laden/Starten vom Regler gelöscht wurde.

Wenn die Daten aufgrund eines Fehlers in der Befehlsgruppe nicht erfolgreich geladen werden, lädt der Antrieb eine Fehlerantwort in die Antwortgruppe (Antworttyp = 0x14, Byte 4 = Fehlercode, Byte 5 = Zusatzcode, Bytes 6-7 Echobefehlsgruppenbytes 2-3). Für weitere Informationen siehe Abgerufene E/A Antwortgruppe 0x14 – Befehl-/Antwortfehler.

Abgerufene E/A-Handshaking-Sequenz	Beispiel
1. Regler lädt einen gültigen Befehlstyp und Daten in die Befehlsgruppe mit Laden/Starten auf Niedrig (0).	Laden Sie einen Zielpositionsbefehl von 1000. C: 0x80 0x00 0x21 0x20 0xE8 0x03 0x00 0x00 Aktivieren=1, Laden/Starten=0, Befehlsachse=1, Befehlstyp=1, Antwortachse=1, Antworttyp=0 (keiner), Daten=1000
2. Antrieb löscht das Flag “Laden abgeschlossen” in der Befehlsgruppe, wenn Laden/Starten in der Befehlsgruppe Niedrig ist.	Antworten Sie mit Statusflags. Noch kein Befehl. R: 0x84 0x00 0x00 0x20 0x00 0x00 0x00 0x00 Aktiviert=1, In Position=1, Laden abgeschlossen=0, Antwortachse=1, Antworttyp=0 (keiner), Daten=0
3. Regler prüft, ob das Flag “Laden abgeschlossen” in der Befehlsgruppe Niedrig ist, um sicherzustellen, dass der Antrieb für den Empfang von Daten bereit ist. Regler setzt das Flag “Daten laden” in der Befehlsgruppe.	Setzen Sie das Flag “Daten laden”. C: 0x81 0x00 0x21 0x20 0xE8 0x03 0x00 0x00 Aktivieren=1, Laden/Starten=1, Befehlsachse=1, Befehlstyp=1, Antwortachse=1, Daten=1000
4. Antrieb sieht Flagwechsel “Laden/Starten” Hoch und versucht, den im Feld Befehlstyp angegebenen Befehl auf den Daten in den Datenbytes auszuführen. Wenn dies erfolgreich ist, setzt der Antrieb das Flag “Laden abgeschlossen”. Wenn der Befehl fehlschlägt oder die Befehlsgruppe ungültig ist, setzt der Antrieb den Antworttyp auf Fehler und lädt die Fehlerinformation in die Antwortgruppendatenfelder. Wenn der Befehl der Betriebsart entspricht (z.B. Zielposition in Positionierungsmodus), beginnt der Antrieb die Bewegung.	Wenn kein Fehler auftritt, führen Sie den angeforderten Befehl aus. R: 0x81 0x00 0x80 0x20 0x00 0x00 0x00 0x00 Aktiviert=1, In Bewegung=1, Laden abgeschlossen=1, Antwortachse=1, Antworttyp=0 (keiner), Daten=0 Im Fehlerfall (z.B. Daten außerhalb des Bereichs): R: 0x80 0x00 0x00 0x34 0x09 0xFF 0x21 0x20 Aktiviert=1, Laden abgeschlossen=0, Antwortachse=1, Antworttyp=0x14 (Fehler), Fehlercodes=0x09FF (Ungültiges Attribut), Bytes 6-7 Echo 2-3.
5. Regler wartet, bis entweder das Flag “Laden abgeschlossen” auf Hoch wechselt oder ein Fehlerantworttyp in der Antwortgruppe aufscheint und löscht dann Laden/Starten. Bereit für den nächsten Befehl.	Löschen Sie Laden/Starten. C: 0x80 0x00 0x21 0x20 0xE8 0x03 0x00 0x00 Aktivieren=1, Laden/Starten=0, Befehlsachse=1, Befehlstyp=1, Antwortachse=1, Daten=1000

5.1.4 Befehlsgruppe 0x01 – Zielposition

Mit dieser Befehlsgruppe wird eine Trajektorie (nur Positionsmodus) des festgelegten Abstands gestartet.
Die Trajektorie kann absolut oder relativ sein, abhängig vom Wert des Bits Relativ. Im Positionsmodus beginnt eine Bewegung, sobald die Zielposition geladen ist. Laden Sie die Zielposition daher zuletzt und erst nach Geschwindigkeit, Beschleunigung und Geschwindigkeitsabnahme.
Die hier gespeicherte Zielposition entspricht dem Zielpositionsattribut des Objekts "Lageregler" (Klasse 0x25, Instanz 1, Attribut 0x06). Auf den Wert kann auch über den seriellen Terminalbefehl O_P zugegriffen werden. Diese Gruppe beeinflusst ausschliesslich Fahrauftrag 0, die Blockbefehle (Fahrsätze) 1 bis 255 bleiben unverändert.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Aktivieren	Reg. aktivieren	Sofortiger Stopp	Kontrollierter Stopp	Richtung	Inkrementell	Startblock	Laden / Start
1	Blocknummer							
2	Befehlsachse = 001			Typ der Eingangsbefehlsgruppe (00001)				
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe				
4	Zielposition niederwertiges Byte							
5	Zielposition niederwertiges, mittleres Byte							
6	Zielposition höherwertiges, mittleres Byte							
7	Zielposition höherwertiges Byte							

Für Bit-Beschreibungen siehe Steuer-Bits und Datenfelder.

Im Folgenden ist ein Beispiel für die Antriebe SERVOSTAR dargestellt. Die Zielposition ist auf 1000 Positionseinheiten oder 0x000003E8 hex eingestellt. Das Bit Aktivieren ist eingestellt, da diese Einstellung die Bewegung initiiert; das Bit Relativ ist eingestellt, um eine relative Positionsbewegung anzugeben; das Bit Laden/Starten ist eingestellt, um mit dem Handshaking zu beginnen. In diesem Beispiel ist der Antwortgruppentyp auf Antwortgruppe 3 – Ist-Geschwindigkeit eingestellt. Der Antrieb überträgt die Ist-Geschwindigkeit, wenn er auf diesen Befehl antwortet.

Für Details zum Senden eines Abgerufene E/A-Befehls siehe den obigen Abschnitt "Data Handshaking". Befolgen Sie die korrekte Sequenz mit dem Befehlsgruppen-Bit Laden/Starten und dem Antwortgruppen-Bit Laden abgeschlossen. Prüfen Sie auf Fehlerantworten, indem Sie Fehlerantwortcode 0x14 im Feld Antwortgruppentyp der Antwortgruppe suchen.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	0	0	1	0	1
1	0							
2	0	0	1	0	0	0	0	1
3	0	0	1	0	0	0	1	1
4	0xE8							
5	0x03							
6	0x00							
7	0x00							

5.1.5 Befehlsgruppe 0x02 – Zielgeschwindigkeit

Diese Befehlsgruppe dient zur Änderung der Zielgeschwindigkeit in Positions- oder Geschwindigkeitsmodus. Das Bit “Richtung“ setzt die gewünschte Richtung im Geschwindigkeitsmodus und wird in allen anderen Modi ignoriert. Im Geschwindigkeitsmodus beginnt eine Bewegung, sobald die Zielgeschwindigkeit geladen ist. Im Positionsmodus beginnt keine Bewegung, wenn die Zielgeschwindigkeit geladen ist.

Im Lagereglermodus entspricht die hier gespeicherte Zielgeschwindigkeit dem Zielgeschwindigkeitsattribut des Objekts “Lageregler” (Klasse 0x25, Instanz 1, Attribut 0x07). Im Drehzahlreglermodus entspricht die hier gespeicherte Zielgeschwindigkeit dem Tippgeschwindigkeitsattribut des Objekts “Lageregler” (Klasse 0x25, Instanz 1, Attribut 0x16). Diese Gruppe beeinflusst ausschliesslich Fahrauftrag 0, die Blockbefehle (Fahrsätze) 1 bis 255 bleiben unverändert.

Die Einheiten werden durch die Antriebseinrichtung (VUNIT, Lageregler-Attribute 40-41) festgelegt.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Aktivieren	Reg. aktivieren	Sofortiger Stopp	Kontrollierter Stopp	Richtung	Inkrementell	Startblock	Laden / Start
1	Blocknummer							
2	Befehlsachse = 001			Typ der Eingangsbefehlsgruppe (00010)				
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe				
4	Zielgeschwindigkeit niederwertiges Byte							
5	Zielgeschwindigkeit niederwertiges, mittleres Byte							
6	Zielgeschwindigkeit höherwertiges, mittleres Byte							
7	Zielgeschwindigkeit höherwertiges Byte							

Für Bit-Beschreibungen siehe Steuer-Bits und Datenfelder.

Im Folgenden ist ein Beispiel für die Antriebe SERVOSTAR dargestellt. Die Zielgeschwindigkeit ist auf 20000 Schritte pro Sekunde oder 0x00004e20 hex eingestellt. Das Bit Aktivieren wird zum Aktivieren des Antriebs eingestellt; die Bit Richtung wird gelöscht, so dass die Bewegung in negativer Richtung erfolgt, wenn der Antrieb in Geschwindigkeitsmodus ist; das Bit Laden/Starten ist eingestellt, um das Handshaking zu beginnen. Im Geschwindigkeitsmodus beschleunigt der Antrieb sofort auf -20000 Schritte pro Sekunde oder verringert die Geschwindigkeit um diesen Wert. Im Positionsmodus wird die Zielgeschwindigkeit für die Trajektorie geladen. In diesem Beispiel ist der Antwortgruppentyp auf Antwortgruppe 1 – Ist-Position eingestellt. Der Antrieb überträgt die Ist-Position, wenn er auf diesen Befehl antwortet (für weitere Informationen siehe Antwortgruppe 0x01 – Ist-Position).

Für Details zum Senden eines Abgerufene E/A-Befehls siehe den obigen Abschnitt “Data Handshaking”. Befolgen Sie die korrekte Sequenz mit dem Befehlsgruppen-Bit Laden/Starten und dem Antwortgruppen-Bit Laden abgeschlossen. Prüfen Sie auf Fehlerantworten, indem Sie Fehlerantwortcode 0x14 im Feld Antwortgruppentyp der Antwortgruppe suchen.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	0	0	0	0	1
1	0							
2	0	0	1	0	0	0	1	0
3	0	0	1	0	0	0	0	1
4	0x20							
5	0x4e							
6	0x00							
7	0x00							

5.1.6 Befehlsgruppe 0x03 – Beschleunigung

Diese Befehlsgruppe dient zur Änderung der Beschleunigung in Positions- oder Geschwindigkeitsmodus.

Der hier gespeicherte Beschleunigungswert entspricht dem Beschleunigungsattribut des Objekts “Lageregler” (Klasse 0x25, Instanz 1, Attribut 0x08). Im Geschwindigkeitsmodus ist der Antriebsparameter ACC eingestellt. Im Positionsmodus ist der Antriebsparameter ACCR eingestellt.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Aktivieren	Reg. aktivieren	Sofortiger Stopp	Kontrollierter Stopp	Richtung	Inkrementell	Startblock	Laden / Start
1	Blocknummer							
2	Befehlsachse = 001			Typ der Eingangsbefehlsgruppe (00011)				
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe				
4	Beschleunigung niederwertiges Byte							
5	Beschleunigung niederwertiges, mittleres Byte							
6	Beschleunigung höherwertiges, mittleres Byte							
7	Beschleunigung höherwertiges Byte							

Für Bit-Beschreibungen siehe Steuer-Bits und Datenfelder.

Im Folgenden ist ein Beispiel für die Antriebe SERVOSTAR dargestellt. Die Beschleunigung ist auf 20000 Schritte/Sek^2 oder 0x00004e20 hex eingestellt. Das Bit Aktivieren ist eingestellt, um den Antrieb zu aktivieren; das Bit Laden/Starten ist eingestellt, um mit dem Handshaking zu beginnen. In diesem Beispiel ist der Antwortgruppentyp auf Antwortgruppe 1 – Ist-Position eingestellt. Der Antrieb überträgt die Ist-Position, wenn er auf diesen Befehl antwortet (für weitere Informationen siehe Antwortgruppe 0x01 – Ist-Position).

Für Details zum Senden eines Abgerufene E/A-Befehls siehe den obigen Abschnitt “Data Handshaking”. Befolgen Sie die korrekte Sequenz mit dem Befehlsgruppen-Bit Laden/Starten und dem Antwortgruppen-Bit Laden abgeschlossen. Prüfen Sie auf Fehlerantworten, indem Sie Fehlerantwortcode 0x14 im Feld Antwortgruppentyp der Antwortgruppe suchen.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	0	0	0	0	1
1	0							
2	0	0	1	0	0	0	1	1
3	0	0	1	0	0	0	0	1
4	0x20							
5	0x4e							
6	0x00							
7	0x00							

5.1.7 Befehlsgruppe 0x04 – Verzögerung

Diese Befehlsgruppe dient zur Änderung der Geschwindigkeitsabnahme in Positions- oder Geschwindigkeitsmodus.

Der hier gespeicherte Wert für die Geschwindigkeitsabnahme entspricht dem Geschwindigkeitsabnahmeattribut des Objekts “Lageregler” (Klasse 0x25, Instanz 1, Attribut 0x09). Im Geschwindigkeitsmodus ist der Antriebsparameter DEC eingestellt. Im Positionsmodus ist der Antriebsparameter DECR eingestellt.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Aktivieren	Reg. aktivieren	Sofortiger Stopp	Kontrollierter Stopp	Richtung	Inkrementell	Startblock	Laden / Start
1	Blocknummer							
2	Befehlsachse = 001			Typ der Eingangsbefehlsgruppe (00100)				
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe				
4	Verzögerung niederwertiges Byte							
5	Verzögerung niederwertiges, mittleres Byte							
6	Verzögerung höherwertiges, mittleres Byte							
7	Verzögerung höherwertiges Byte							

Für Bit-Beschreibungen siehe Steuer-Bits und Datenfelder.

Im Folgenden ist ein Beispiel für die Antriebe SERVOSTAR dargestellt. Die Geschwindigkeitsabnahme ist auf 20000 Schritte/Sek^2 oder 0x00004e20 hex eingestellt. Das Bit Aktivieren ist eingestellt, um den Antrieb zu aktivieren; das Bit Laden/Starten ist eingestellt, um mit dem Handshaking zu beginnen. In diesem Beispiel ist der Antwortgruppentyp auf Antwortgruppe 1 – Ist-Position eingestellt. Der Antrieb überträgt die Ist-Position, wenn er auf diesen Befehl antwortet (für weitere Informationen siehe Antwortgruppe 0x01 – Ist-Position).

Für Details zum Senden eines Abgerufene E/A-Befehls siehe den obigen Abschnitt “Data Handshaking”. Befolgen Sie die korrekte Sequenz mit dem Befehlsgruppen-Bit Laden/Starten und dem Antwortgruppen-Bit Laden abgeschlossen. Prüfen Sie auf Fehlerantworten, indem Sie Fehlerantwortcode 0x14 im Feld Antwortgruppentyp der Antwortgruppe suchen.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	0	0	0	0	1
1	0							
2	0	0	1	0	0	1	0	0
3	0	0	1	0	0	0	0	1
4	0x20							
5	0x4e							
6	0x00							
7	0x00							

5.1.8 Befehlsgruppe 0x05 – Drehmoment

Diese Befehlsgruppe dient zur Änderung des Drehmoments. Diese Befehle können nur im Drehmomentmodus verwendet werden. Die Bewegung beginnt, sobald der Wert geladen ist.

Der hier gespeicherte Drehmomentwert entspricht dem Drehmomentattribut des Objekts “Lageregler” (Klasse 0x25, Instanz 1, Attribut 0x19). Auf den Wert kann auch über den seriellen Terminalbefehl T zugegriffen werden.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Aktivieren	Reg. aktivieren	Sofortiger Stopp	Kontrollierter Stopp	Richtung	Inkrementell	Startblock	Laden / Start
1	Blocknummer							
2	Befehlsachse = 001			Typ der Eingangsbefehlsgruppe (00101)				
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe				
4	Drehmoment niederwertiges Byte							
5	Drehmoment niederwertiges, mittleres Byte							
6	Drehmoment höherwertiges, mittleres Byte							
7	Drehmoment höherwertiges Byte							

Für Bit-Beschreibungen siehe Steuer-Bits und Datenfelder.

Im Folgenden ist ein Beispiel für die Antriebe SERVOSTAR dargestellt. In diesem Beispiel ist das Drehmoment (Strom) auf 3,0 A in einem Antrieb mit einem Spitzenwert von 6,0 A gesetzt. Die Einheiten für das Drehmoment sind auf einen Spitzenstrom von 3280 gewichtet, sodass der Sollwert 3280*3,0/6,0=1640 Drehmomenteinheiten (hexadezimal 0x00000668 Drehmomenteinheiten) beträgt. Das Bit Aktivieren ist eingestellt, um den Antrieb zu aktivieren; das Bit Laden/Starten ist eingestellt, um mit dem Handshaking zu beginnen. Im Drehmomentmodus beginnt die Bewegung, sobald der Drehmomentbefehl geladen ist. In diesem Beispiel ist der Antwortgruppentyp auf Antwortgruppe 1 – Ist-Position eingestellt. Der Antrieb überträgt die Ist-Position, wenn er auf diesen Befehl antwortet (für weitere Informationen siehe Antwortgruppe 0x01 – Ist-Position).

Für Details zum Senden eines Abgerufene E/A-Befehls siehe den obigen Abschnitt “Data Handshaking”. Befolgen Sie die korrekte Sequenz mit dem Befehlsgruppen-Bit Laden/Starten und dem Antwortgruppen-Bit Laden abgeschlossen. Prüfen Sie auf Fehlerantworten, indem Sie Fehlerantwortcode 0x14 im Feld Antwortgruppentyp der Antwortgruppe suchen.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	0	0	0	0	1
1	0							
2	0	0	1	0	0	1	0	1
3	0	0	1	0	0	0	0	1
4	0x68							
5	0x06							
6	0x00							
7	0x00							

5.2 E/A-Antwortgruppen

Abgerufene E/A Messaging ist eine Methode zum Übertragen einer Gruppe von Steuer-Bits und eines Datenbefehls und erhalten einer Gruppe von Status-Bits und einer Antwort mit einem Datenwert. Diese Kommunikationsmethode ist bevorzugt, da mit Explicit Messaging nur jeweils ein Wert übertragen werden kann. Abgerufene E/A-Messaging ist eine Methode zur Übermittlung einer Gruppe spezifischer Befehle an Geräte. Diese Kommunikationsmethode ist vorzuziehen, da sie schneller ist als Explicit Messaging. Abgerufene E/A und Explicit Messaging können gleichzeitig für die Kommunikation zwischen dem Regler und dem Antrieb verwendet werden. In diesem Abschnitt wird das Format für jede Antwortgruppe definiert. Ausserdem enthält dieser Abschnitt Beispiele für jede Gruppe.

Bei Abgerufene E/A-Messaging überträgt der Antrieb eine Antwortgruppe, wenn er eine Befehlsgruppe vom Antrieb erhält. Die Antwortgruppe enthält Steuer-Bits, die für jeden Antworttyp identisch definiert sind. Zusätzlich zu den Status-Bits kann eine Antwortgruppe jeweils einen Datenwert übertragen (Ist-Position, befohlene Position, Ist-Geschwindigkeit, Ist-Drehmoment oder Fehlerwerte). Der Antworttyp wird im Feld Antwortgruppentyp der Befehlsgruppe angegeben. Eine Befehlsgruppe kann sowohl einen Befehlsgruppentyp und einen Antwortgruppentyp enthalten, um einen Befehl zu übertragen und einen Datenwert in der selben Gruppe anzufordern.

5.2.1 Status-Bits und Datenfelder

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status aktivieren	Registrierungsebene	Referenzfahrtebene	Aktuelle Richtung	Allgemeiner Fehler	In Position	Block in Ausführung	In Bewegung
1	Ausführung der Blocknummer							
2	Laden beendet	Blockfehler	Folgefehler	Negative SW-Grenze	Positive SW-Grenze	Negative HW-Grenze	Positive HW-Grenze	Fehlereingang aktiv
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe				
4	Daten Niederwertiges Byte							
5	Daten Niederwertiges, mittleres Byte							
6	Daten Höherwertiges, mittleres Byte							
7	Daten Höherwertiges Byte							

Status aktivieren Dieses Bit gibt den Aktivierungsstatus des Antriebs wieder. Siehe "Aktivieren (Klasse 37: Lageregler, Attribut 17)".

Registrierungsebene Die tatsächliche Stufe des Registrierungseingangs. Der digitale Eingang 2 muss für diese Registrierung konfiguriert sein.

Referenzfahrtebene Dieses Bit gibt die Ebene des Referenzfahrteingangs des Antriebs wieder.

Aktuelle Richtung Dieses Bit gibt die tatsächliche Bewegungsrichtung an. Wenn der Antrieb nicht in Bewegung ist, wird die Richtung der letzten Bewegung angezeigt. Siehe auch “Richtung” (Klasse 0x25, Attribut 0x17).

Allgemeiner Fehler Dieses Bit zeigt an, ob ein Fehler aufgetreten ist. Siehe "Allgemeiner Fehler (Klasse 36: Lageregler Überwachung, Attribut 5)".

In Position Dieses Bit gibt an, ob sich der Motor in der zuletzt angestrebten Position (1-Im Ziel) befindet. Siehe Beschreibung für "Relative Position (Klasse 37: Lageregler, Attribut 12)".

Block in Ausführung Wenn dieses Bit gesetzt ist, führt der Antrieb ein Blockbefehlprogramm durch. Siehe auch “Block ausführen” (Klasse 0x26 “Blockfolgesteuerung”, Attribut 0x02).

In Bewegung Dieses Bit gibt an, ob eine Trajektorie abläuft (1) oder beendet (0) ist. Das Bit wird sofort für die Befehlsgruppen 1, 10 und 11 gesetzt und bleibt während der gesamten Bewegung gesetzt. Siehe Beschreibung für "Trajektoriestart (Klasse 37 - Lageregler, Attribut 11)".

Laden beendet Dieses Bit zeig an, dass die Befehlsdaten in der Befehlsmeldung

erfolgreich in den Verstärker geladen wurden. Wird für das handshaking zwischen Steuerung und Verstärker verwendet, siehe auch Data Handshaking.

Blockfehler
Dieses Bit wird gesetzt, um anzuzeigen, dass in der Befehlsblockprogrammsequenz ein Fehler aufgetreten ist. Lesen Sie den Blockfehlercode (Klasse 0x28 Blockfolgesteuerung, Attribut 0x05), um den Fehler zu beseitigen. Siehe auch Blockfehler (Klasse 0x26 “Blockfolgesteuerung”, Attribut 0x04).

Folgefehler
Dieses Bit zeigt an, wann ein Folgefehler (statisch oder dynamisch) auftritt. Beseitigen Sie den Fehler, um mit der Bewegung fortzufahren (Klasse 0x24 Lagereglerüberwachung, Attribut 0x64 – Fehler löschen).

Negative SW Grenze
Dieses Bit zeigt an, wann die Position unter dem oder am negativen Softwareendschalter ist.

Positive SW Grenze
Dieses Bit zeigt an, wann die Position unter dem oder am positiven Softwareendschalter ist.

Negative HW Grenze
Dieses Bit zeigt den Status für den Endschaltereingang bei Linksdrehung an.

Positive HW Grenze
Dieses Bit zeigt den Status für den Endschaltereingang bei Rechtsdrehung an.

Fehlereingang aktiv
Dieses Bit zeigt den Status für den Nothalteingang an. Einer der digitalen Eingänge 1 bis 4 muss als Nothalteingang definiert sein.

Antwortachse
Der Antrieb unterstützt nur eine Achse; daher muss dieser Wert immer 1 sein. Der Wert wird von der Befehlsgruppe gespiegelt.

Antwortgruppentyp
Setzen Sie den Antworttyp in der Befehlsgruppe, um festzulegen, welche Daten im Datenfeld der Antwortgruppe zurückgeliefert werden. Die Ist-Position, Position, Ist-Geschwindigkeit und das Ist-Drehmoment sind verfügbar. Die Antwortgruppe spiegelt den Antwortgruppentyp von der Befehlsgruppe, ausgenommen bei einem Fehler in der Befehlsgruppe. Wenn die Befehlsgruppe ungültig ist, wird der Antwortgruppentyp der Antwortgruppe auf 0x14 gesetzt (Fehlerantwort) und ein Fehlercode wird im Datenfeld ausgegeben.

Daten-Bytes
Antwortdaten für den gewünschten Antworttyp werden in die Datenfelder geladen, mit dem niederwertigsten Byte zuerst.

5.2.2 Antwortgruppe 0x01 – Istposition

Mit dieser Antwortgruppe wird die Istposition des Motors (in Positionseinheiten) zurückgegeben.

Die hier gelieferte Ist-Position entspricht dem Attribut der Ist-Position des Objekts “Lageregler” (Klasse 0x25, Instanz 1, Attribut 0x0D). Auf den Wert kann auch über den seriellen Terminalbefehl PFB zugegriffen werden.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status aktivieren	Registrierungsebene	Referenzfahrtebene	Aktuelle Richtung	Allgemeiner Fehler	In Position	Block in Ausführung	In Bewegung
1	Ausführung der Blocknummer							
2	Laden beendet	Blockfehler	Folgefehler	Negative SW-Grenze	Positive SW-Grenze	Negative HW-Grenze	Positive HW-Grenze	Fehlereingang aktiv
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe (00001)				
4	Istposition Niederwertiges Byte							
5	Istposition Niederwertiges, mittleres Byte							
6	Istposition Höherwertiges, mittleres Byte							
7	Istposition Höherwertiges Byte							

Für Definitionen der einzelnen Bits und Felder siehe Status-Bits und Datenfelder.

Im Folgenden ist ein Beispiel für die Antriebe SERVOSTAR dargestellt. Die Ist-Position ist auf 10.000 Positionseinheiten oder 0x00002710 hex eingestellt.

Aktivierungszustand = 1 (aktiviert)
Registrierungsebene = 0 (nicht aktiv)
Referenzfahrtebene = 0 (nicht auf dem Referenzfahrt-Flag)
Aktuelle Richtung = 1 (positive Richtung)
Allgemeine Fehler = 0 (keine Fehler)
In der Zielposition = 1 (in Position)
Blockausführung = 0 (Befehlsblockprogramm wird nicht ausgeführt)
In Bewegung = 0 (nicht in Bewegung)
Laden abgeschlossen = 1 (Befehlsgruppendaten erfolgreich geladen)
Blockfehler = 0 (kein Fehler)
Folgefehler = 0 (kein Fehler)
Negatives SW-Limit = 0 (nicht erreicht)
Positives SW-Limit = 1 (erreicht)
Negatives HW-Limit = 0 (Schalter für negatives Richtungslimit inaktiv)
Positives HW-Limit = 1 (Schalter für positives Richtungslimit aktiv)
Fehlereingang aktiv = 0 (Notstoppeingänge inaktiv)
Antwortachse = 001
Antwortgruppentyp = 00001
Daten = 0x00002710

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	1	0	1	0	1
1	0							
2	1	0	0	0	1	0	1	0
3	0	0	1	0	0	0	0	1
4	0x10							
5	0x27							
6	0x01							
7	0x00							

5.2.3 Antwortgruppe 0x02 – Befohlene Lage

Mit dieser Antwortgruppe wird die befohlene Lage des Motors (in Positionseinheiten) zurückgegeben.

Auf den Wert kann auch über den seriellen Terminalbefehl PTARGET zugegriffen werden.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status aktivieren	Registrierungsebene	Referenzfahrtebene	Aktuelle Richtung	Allgemeiner Fehler	In Position	Block in Ausführung	In Bewegung
1	Ausführung der Blocknummer							
2	Laden beendet	Blockfehler	Folgefehler	Negative SW-Grenze	Positive SW-Grenze	Negative HW-Grenze	Positive HW-Grenze	Fehlereingang aktiv
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe (00010)				
4	Befohlene Lage Niederwertiges Byte							
5	Befohlene Lage Niederwertiges, mittleres Byte							
6	Befohlene Lage Höherwertiges, mittleres Byte							
7	Befohlene Lage Höherwertiges Byte							

Für Definitionen der einzelnen Bits und Felder siehe Status-Bits und Datenfelder.

Im Folgenden ist ein Beispiel für die Antriebe SERVOSTAR dargestellt. Die befohlene Position ist auf 10.000 Positionseinheiten oder 0x00002710 hex eingestellt.

Aktivierungszustand = 1 (aktiviert)
Registrierungsebene = 0 (nicht aktiv)
Referenzfahrtebene = 0 (nicht auf dem Referenzfahrt-Flag)
Aktuelle Richtung = 1 (positive Richtung)
Allgemeine Fehler = 0 (keine Fehler)
In der Zielposition = 1 (in Position)
Blockausführung = 0 (Befehlsblockprogramm wird nicht ausgeführt)
In Bewegung = 0 (nicht in Bewegung)
Laden abgeschlossen = 1 (Befehlsgruppendaten erfolgreich geladen)
Blockfehler = 0 (kein Fehler)
Folgefehler = 0 (kein Fehler)
Negatives SW-Limit = 0 (nicht erreicht)
Positives SW-Limit = 1 (erreicht)
Negatives HW-Limit = 0 (Schalter für negatives Richtungslimit inaktiv)
Positives HW-Limit = 0 (Schalter für positives Richtungslimit inaktiv)
Fehlereingang aktiv = 0 (Notstoppeingänge inaktiv)
Antwortachse = 001
Antwortgruppentyp = 00010
Daten = 0x00002710

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	1	0	1	0	1
1	0							
2	1	0	0	0	1	0	0	0
3	0	0	1	0	0	0	1	0
4	0x10							
5	0x27							
6	0x00							
7	0x00							

5.2.4 Antwortgruppe 0x03 – Istgeschwindigkeit

Diese Antwortgruppe gibt die Istgeschwindigkeit des Motors (in Positionseinheiten/s) wieder. Die Einheiten werden durch die Antriebseinrichtung (VUNIT, Lageregler-Attribute 40-41) festgelegt.

Die hier gelieferte Ist-Geschwindigkeit entspricht dem Attribut der Ist-Geschwindigkeit des Objekts “Lageregler” (Klasse 0x25, Instanz 1, Attribut 0x0E). Auf den Wert kann auch über den seriellen Terminalbefehl PV zugegriffen werden.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status aktivieren	Registrierungsebene	Referenzfahrtebene	Aktuelle Richtung	Allgemeiner Fehler	In Position	Block in Ausführung	In Bewegung
1	Ausführung der Blocknummer							
2	Laden beendet	Blockfehler	Folgefehler	Negative SW-Grenze	Positive SW-Grenze	Negative HW-Grenze	Positive HW-Grenze	Fehlereingang aktiv
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe (00011)				
4	Istgeschwindigkeit Niederwertiges Byte							
5	Istgeschwindigkeit Niederwertiges, mittleres Byte							
6	Istgeschwindigkeit Höherwertiges, mittleres Byte							
7	Istgeschwindigkeit Höherwertiges Byte							

Für Definitionen der einzelnen Bits und Felder siehe Status-Bits und Datenfelder.

Im Folgenden ist ein Beispiel für die Antriebe SERVOSTAR dargestellt. Die Ist-Geschwindigkeit ist auf 10.000 Positionseinheiten/Sekunde oder 0x00002710 hex eingestellt.

Aktivierungszustand = 1 (aktiviert)
Registrierungsebene = 0 (nicht aktiv)
Referenzfahrtebene = 0 (nicht auf dem Referenzfahrt-Flag)
Aktuelle Richtung = 1 (positive Richtung)
Allgemeine Fehler = 0 (keine Fehler)
In der Zielposition = 1 (in Position)
Blockausführung = 0 (Befehlsblockprogramm wird nicht ausgeführt)
In Bewegung = 0 (nicht in Bewegung)
Laden abgeschlossen = 1 (Befehlsgruppendaten erfolgreich geladen)
Blockfehler = 0 (kein Fehler)
Folgefehler = 0 (kein Fehler)
Negatives SW-Limit = 0 (nicht erreicht)
Positives SW-Limit = 1 (erreicht)
Negatives HW-Limit = 0 (Schalter für negatives Richtungslimit inaktiv)
Positives HW-Limit = 0 (Schalter für positives Richtungslimit inaktiv)
Fehlereingang aktiv = 0 (Notstoppeingänge inaktiv)
Antwortachse = 001
Antwortgruppentyp = 00011
Daten = 0x00002710

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	1	0	1	0	1
1	0							
2	1	0	0	0	1	0	0	0
3	0	0	1	0	0	0	1	1
4	0x10							
5	0x27							
6	0x00							
7	0x00							

5.2.5 Antwortgruppe 0x05 – Drehmoment

Diese Antwortgruppe gibt das Istdrehmoment (Strom) des Motors wieder.

Das hier gelieferte Ist-Drehmoment entspricht dem Drehmomentattribut des Objekts “Lageregler” (Klasse 0x25, Instanz 1, Attribut 0x19). Auf den Wert kann auch über den seriellen Terminalbefehl T zugegriffen werden.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status aktivieren	Registrierungsebene	Referenzfahrtebene	Aktuelle Richtung	Allgemeiner Fehler	In Position	Block in Ausführung	In Bewegung
1	Ausführung der Blocknummer							
2	Laden beendet	Blockfehler	Folgefehler	Negative SW-Grenze	Positive SW-Grenze	Negative HW-Grenze	Positive HW-Grenze	Fehlereingang aktiv
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe (00101)				
4	Drehmoment Niederwertiges Byte							
5	Drehmoment Niederwertiges, mittleres Byte							
6	Drehmoment Höherwertiges, mittleres Byte							
7	Drehmoment Höherwertiges Byte							

Für Definitionen der einzelnen Bits und Felder siehe Status-Bits und Datenfelder.

Im Folgenden ist ein Beispiel für die Antriebe SERVOSTAR dargestellt. Das Istdrehmoment (Strom) beträgt 3,0 A für einen Antrieb mit einem Spitzenwert von 6,0 A. Die Einheiten für das Drehmoment sind auf einen Spitzenstrom von 3280 gewichtet, so dass der Ist-Drehmomentwert 3280*3,0/6,0=1640 Drehmomenteinheiten (hexadezimal 0x00000668 Drehmomenteinheiten) beträgt.

Aktivierungszustand = 1 (aktiviert)
Registrierungsebene = 0 (nicht aktiv)
Referenzfahrtebene = 0 (nicht auf dem Referenzfahrt-Flag)
Aktuelle Richtung = 1 (positive Richtung)
Allgemeine Fehler = 0 (keine Fehler)
In der Zielposition = 1 (in Position)
Blockausführung = 0 (Befehlsblockprogramm wird nicht ausgeführt)
In Bewegung = 1 (in Bewegung)
Laden abgeschlossen = 0 (keine Daten von der Befehlsgruppe geladen)
Blockfehler = 0 (kein Fehler)
Folgefehler = 0 (kein Fehler)
Negatives SW-Limit = 0 (nicht erreicht)
Positives SW-Limit = 0 (nicht erreicht)
Negatives HW-Limit = 0 (Schalter für negatives Richtungslimit inaktiv)
Positives HW-Limit = 0 (Schalter für positives Richtungslimit inaktiv)
Fehlereingang aktiv = 0 (Notstoppeingänge inaktiv)
Antwortachse = 001
Antwortgruppentyp = 00101
Daten = 0x00000668

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	1	0	1	0	1
1	0							
2	0	0	0	0	1	0	0	0
3	0	0	1	0	0	1	0	1
4	0x68							
5	0x06							
6	0x00							
7	0x00							

5.2.6 Antwortgruppe 0x14 – Befehl-/Antwortfehler

Diese Antwort identifiziert einen aufgetretenen Fehler. Diese Antwort wird immer als Reaktion auf eine ungültige Befehlsgruppe ausgegeben. Das Feld Antwortgruppentyp der Antwortgruppe spiegelt normalerweise das entsprechende Feld aus der vorigen Befehlsgruppe wider. Im Fall einer ungültigen Befehlsgruppe wird das Feld Antwortgruppentyp der Antwortgruppe auf 0x14 gesetzt und Fehlercodes werden im Datenfeld ausgegeben.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Status aktivieren	Registrierungsebene	Referenzfahrtebene	Aktuelle Richtung	Allgemeiner Fehler	In Position	Block in Ausführung	In Bewegung
1	Ausführung der Blocknummer							
2	Laden beendet	Blockfehler	Folgefehler	Negative SW-Grenze	Positive SW-Grenze	Negative HW-Grenze	Positive HW-Grenze	Fehlereingang aktiv
3	Antwortachse = 001			Typ der Ausgangsantwortgruppe (10100)				
4	Allgemeiner Fehlercode							
5	Zusätzlicher Code							
6	Kopie der Befehlsmeldung Byte 2							
7	Kopie der Befehlsmeldung Byte 3							

Fehlercode (hex)	Zusätzlicher Code (hex)	DeviceNet-Fehler
0	FF	NO ERROR
2	FF	RESOURCE_UNAVAILABLE
5	FF	PATH_UNKNOWN
5	1	COMMAND_AXIS_INVALID
5	2	RESPONSE_AXIS_INVALID
8	FF	SERVICE_NOT_SUPP
8	1	COMMAND_NOT_SUPPORTED
8	2	RESPONSE_NOT_SUPPORTED
9	FF	INVALID_ATTRIBUTE_VALUE
B	FF	ALREADY_IN_STATE
C	FF	OBJ_STATE_CONFLICT
D	FF	OBJECT_ALREADY_EXISTS
E	FF	ATTRIBUTE_NOT_SETTABLE
F	FF	ACCESS_DENIED
10	FF	DEVICE_STATE_CONFLICT
11	FF	REPLY_DATA_TOO_LARGE
13	FF	NOT_ENOUGH_DATA
14	FF	ATTRIBUTE_NOT_SUPP
15	FF	TOO_MUCH_DATA
16	FF	OBJECT_DOES_NOT_EXIST
17	FF	FRAGMENTATION_SEQ_ERR
20	FF	INVALID_PARAMETER

Im Folgenden ist ein Beispiel für die Antriebe SERVOSTAR dargestellt. Für die vorangegangene Befehlsgruppe waren der Befehl 0x06 (nicht unterstützt) und die Antwort 0x01 erforderlich. Der Antrieb gibt Antwortgruppe 0x14 (Befehls-/Antwortfehler) mit dem allgemeinen Fehler = 0x08 und zusätzlichen Code = 0x01 (COMMAND_NOT_SUPPORTED) zurück. Die Bytes 2 und 3 aus der Befehlsgruppe werden in der Fehlerantwortgruppe gespiegelt.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	0	1	0	1	0	1
1	0							
2	0	0	0	0	0	0	0	0
3	0	0	1	1	0	1	0	0
4	0x08							
5	0x01							
6	0x26							
7	0x21							

6 Anhang

6.1 DeviceNet SPS-Beispiele

6.1.1 Übersicht

DeviceNet bietet drei Methoden der Bewegungssteuerung – Explizite Meldungen, Gruppenmeldungen und Befehlsblockfolgesteuerungen. Jede Methode hat in bestimmten Situationen gewisse Vorteile gegenüber den anderen Methoden. Explizite Meldungen werden normalerweise für die Konfiguration des Antriebs verwendet. Gruppenmeldungen werden für die Bewegungssteuerung verwendet, und Befehlsblöcke kommen zur Anwendung, wenn gespeicherte Bewegungssequenzen ohne übermäßige SPS-Involvierung ausgeführt werden können.

Die meisten SPSs unterstützen sowohl explizite Meldungen als auch Gruppenmeldungen (Abgerufene E/A) gleichzeitig. Die zuvor beschriebenen Objekte werden allesamt über Explicit Messaging aufgerufen. In diesem Abschnitt finden Sie eine Beschreibung der Verwendung von Befehls- und Antwortgruppen. Befehlsblockfolgesteuerungen können über explizite Meldungen eingerichtet und dann entweder über explizite Meldungen oder Gruppenmeldungen gesteuert werden.

Explizite Meldungen ermöglichen den Zugriff auf jeweils einen einzelnen Parameterwert. Der gewünschte Parameter wird durch die Angabe der Klassennummer, der Instanznummer und der Attributnummer in einer expliziten Meldung ausgewählt. Gruppenmeldungen kombinieren viele Steuer- und Status-Bits in 8-Byte-Befehle und Antwortmeldungen. Sie sind nicht so vielseitig wie explizite Meldungen (nur bestimmte Parameter sind zugreifbar), aber es können mehrere Steuerwerte innerhalb einer Meldung geändert werden. Daher eignen sich explizite Meldungen besser für die Konfiguration und Gruppenmeldungen besser zur Bewegungssteuerung.

Die meisten Antriebskonfigurationen erfolgen innerhalb des Lageregler-Objekts, das die meisten für die Bewegungssteuerung notwendigen Parameter umfasst. Modifizieren Sie Parameter in diesem Objekt, um die Betriebsart (Drehmoment, Geschwindigkeit, Position) einzustellen und Bewegung im SERVOSTAR zu konfigurieren. Zeigen Sie Parameter in diesem Objekt an, um die Antriebsparameter und Statuswörter zu lesen. Eine zusätzliche Antriebskonfiguration kann über ein weiteres unterstütztes Objekt, das Parameterobjekt, erfolgen. Dabei handelt es sich um ein vom Lieferanten definiertes Objekt, das Lieferantenkonfigurationsparameter umfasst. Alle Antriebsparameter mit einer DPR-Nummer von weniger als 256 (siehe ascii.chm-Referenz) können über das Parameterobjekt aufgerufen werden.

Abgerufene E/A-Gruppenmeldungen werden für die meisten Bewegungssteuerungen verwendet. Abgerufene E/A bestehen aus einer Befehlsgruppe vom SPS zum SERVOSTAR und einer Antwortgruppe vom SERVOSTAR zum SPS. Steuer-Bits in einer Befehlsmeldung werden zur Aktivierung des Antriebs, Durchführung eines kontrollierten Stopps des Motors, zum Initiieren von Bewegung und zum Initiieren von gespeicherten Bewegungsblockprogrammen verwendet. Befehlsmeldungen können darüber hinaus die Parameter für Zielposition, Zielgeschwindigkeit, Beschleunigung, Geschwindigkeitsabnahme oder Drehmoment festlegen. Status-Bits in einer Antwortmeldung zeigen Fehlerstati und den allgemeinen Status des Antriebs. Antwortmeldungen können darüber hinaus die Ist-Position, die befohlene Position, die Ist-Geschwindigkeit oder das Drehmoment anzeigen.

Bewegungssequenzen oder Aufgaben können über die Objektklasse “Befehlsblock” vorab in den Antrieb programmiert werden. Diese Blocks entsprechen den SERVOSTAR-Fahraufträgen. Positionierungsbewegungen, Zeitverzögerungen und Parametermodifizierungsblocks können verknüpft werden, um ein im Antrieb gespeichertes Bewegungsblockprogramm zu erstellen. Sobald das gespeicherte Blockprogramm konfiguriert wird, kann es entweder über das Objekt “Blockfolgesteuerung” oder mit dem Blocknummernfeld “Abgerufene E/A Befehlsmeldung” und dem Startblockbit ausgeführt werden.

6.1.2 Antriebseinrichtung für die Beispiele

Zum Testen unserer Beispiele konfigurieren Sie zunächst Ihren Antrieb entsprechend unserer beispieleinstellung:

Auf der Bildschirmseite “Basiseinstellungen” wie folgt:
Beschleunigungseinheiten = ms->VLIM
Geschwindigkeitseinheiten = U/min
Positionseinheiten = Schritte

Auf dem Bildschirm “Position->Lagedaten” wie folgt:
Auflösung = 1000 Schritte/Umdrehung

Auf der Bildschirmseite “I/O digital” wie folgt:
DIGITAL-OUT 1 = 23: Reserviert

(Die digitale Ausgangsfunktion 23 stellt den digitalen Ausgang unter die Kontrolle des Feldbus).

Speichern Sie die Parameter und starten Sie den Antrieb erneut.

In den Beispielen in diesem Abschnitt wird vorausgesetzt, dass diese Einheiten eingerichtet wurden. Der Antrieb muss ebenfalls entsprechend eingestellt und konfiguriert werden, ehe Sie fortfahren können.

Wenn der Antrieb entsprechend konfiguriert ist, führen Sie testweise die folgende Befehlssequenz vom seriellen Terminal aus. Die Motorwelle sollte in 1 Sekunde 1 Umdrehung zurücklegen, wenn der Befehl MOVE 0 eingegeben wird.

Befehl	Beschreibung
OPMODE 8	Einstellen des Lagemodus.
EN	Aktivieren
MH	Referenzfahrt
O_C 8193	Relative Bewegung unter Verwendung von Benutzereinheiten
O_P 1000	Bewegung 1000 Schritte (1 Umdrehung)
O_V 60	Geschwindigkeit=60 U/MIN
O_ACC1 10	Beschleunigung = 10ms auf Zielgeschwindigkeit
O_DEC1 10	Geschwindigkeitsabnahme = 10ms auf Zielgeschwindigkeit
MOVE 0	Ausführen von Fahrsatz 0

6.1.3 Abgerufene E/A-Gruppen

Eine ausführliche Beschreibung von Gruppen finden Sie in den folgenden Abschnitten: “Typische Verwendung von expliziten Meldungen und Gruppenmeldungen”, “Befehlsgruppen” und “Antwortgruppen”. In diesen Abschnitten finden Sie Informationen über die Struktur und die Verwendung von Befehls- und Antwortgruppen.

Für Abgerufene E/A-Gruppenmeldungen initiiert der SPS die Kommunikation mit einer Befehlsgruppe und der Antrieb antwortet mit einer Antwortgruppe. Der SPS sendet diese Meldungen in regelmäßigen “Polling-Abständen”. Über ein Handshaking-Protokoll wird sichergestellt, dass die Daten korrekt übertragen werden – für eine Beschreibung dieses Protokolls siehe den Abschnitt “Data Handshaking”.

6.1.3.1 Senden von Befehlsgruppen – ControlLogix

Der SERVOSTAR kann von Allen Bradleys SPSs der Logix 5000-Reihe mit DeviceNet Scannern (1756-DNB) unter Verwendung von Gruppenmeldungen gesteuert werden. Befehlsgruppen werden vom SPS an den Antrieb gesendet, um die Antriebsbewegung zu steuern. Das Format der Befehlsgruppe wird in Abschnitt 5.1.1 dargestellt.
Für die Kommunikation mit dem SERVOSTAR über Gruppen muss der Antrieb zunächst in der SPS-Scan-Liste abgebildet werden (das Abbildungsverfahren wird in diesem Dokument nicht beschrieben). Zudem muss die Befehlsgruppe im Ausgabespeicher des SPS abgebildet werden. Informationen zur Erzeugung einer Scan-Liste und zur Abbildung des Eingangs- und Ausgangsspeichers mit RSNetworx finden Sie im Handbuch Ihres DeviceNet Scanner. Achten Sie darauf, die richtige Steckplatznummer zu verwenden. Der Befehl und die Antwortgruppen sind jeweils acht Byte lang und werden als je zwei 32-Bit-Wörter im SPS abgebildet. Byte 0-3 der Gruppe werden in Byte 0-3 von Wort 0 und Byte 4-7 der Gruppe werden in Byte 0-3 von Wort 1 abgebildet.
Wenn Sie nur einen einzigen SERVOSTAR in der SPS-Scan-Liste haben und sich der Scanner in Einsteckplatz 1 befindet, kann Ihre Abbildung der Ausgangsdatei etwa so aussehen:

Ausgangswort	Beschreibung
Local:1:O.Data	DN Scanner-Ausgangsspeicher
Local:1:O.Data[0]	DN Befehlswort 0 (Steuerflags, Blocknummer, Befehlsauswahl, Antwortauswahl)
Local:1:O.Data[1]	DN Befehlswort 1 (Befehlsdaten)
Local:1:O.Data[2-123]	Nicht abgebildeter Bereich des DN Scanner (steht für andere Geräte zur Verfügung)

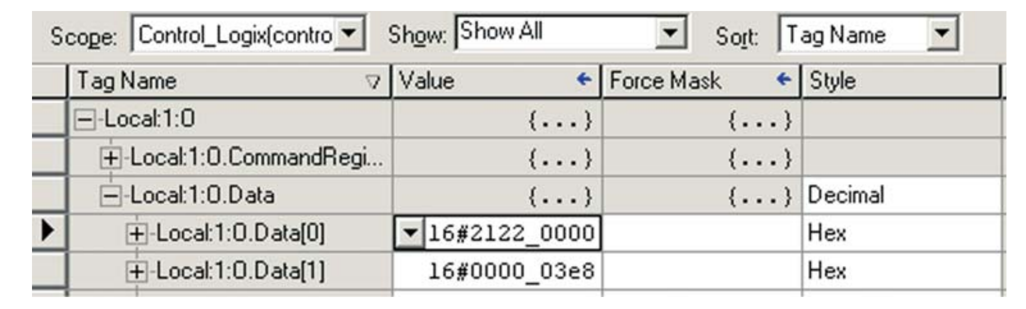
Sobald die Ausgänge abgebildet sind, können Sie die an den Antrieb zu sendende Befehlsgruppe modifizieren, indem Sie in die entsprechenden Ausgangswörter schreiben. Die neuen Daten werden beim nächsten und jedem folgenden Scanzyklus übertragen, bis die Ausgangsdatei wieder modifiziert wird. Setzen Sie die Zielgeschwindigkeit testweise auf 1000. Der Wert kann über das Terminal überprüft werden, indem Sie Befehl O_V eingeben. Ändern Sie die SPS-Ausgangsbefehlsgruppe (Scanner-Ausgangsdaten) wie folgt:

Byte	Funktion	Datenwert (hex)
0	Befehlsflags – Deaktivieren	0x00
1	Blocknummer	0x00
2	Achseninstanz, Befehlsgruppe 2 – Zielgeschwindigkeit	0x22
3	Achseninstanz, Antwortgruppe 1 – Istposition	0x21
4	Zielgeschwindigkeit – niederwertiges Wort niederwertiges Byte	0xE8
5	Zielgeschwindigkeit – niederwertiges Wort höherwertiges Byte	0x03
06	Zielgeschwindigkeit – höherwertiges Wort niederwertiges Byte	0x00
7	Zielgeschwindigkeit – höherwertiges Wort höherwertiges Byte	0x00

Die Abbildung dieser Gruppe im SPS-Speicher sollte etwa so aussehen wie die folgende Tabelle:

Ausgangswort	Datenwert (hex)
Local:1:O.Data[0]	0x2122_0000
Local:1:O.Data[1]	0x0000_03E8

Sie können die Daten testweise direkt von der Bildschirmseite "Programm-Tags" modifizieren:



Der Befehl und die Daten werden nun bei jedem Scanzyklus an den Antrieb übertragen, doch das Data Handshaking-Protokoll muss zum Laden der Daten verwendet werden (weitere Informationen dazu finden Sie bei Befehlsgruppen, Unterabschnitt "Data Handshaking"). Warten Sie, bis die Befehlsgruppe übertragen wurde. Setzen Sie dann das Lade-Bit auf Hoch, indem Sie 0x2122_0001 in Befehlswort 0 (Local:1:O.Data[0]) schreiben. Verwenden Sie nun das serielle Terminal zum Lesen des Werts von O_V – dieser sollte gleich 1000 sein.

6.1.3.2 Lesen von Antwortgruppen – ControlLogix

Eine Antwortgruppe ist eine Abgerufene E/A-Meldung, die vom Antrieb als Antwort auf eine Befehlsgruppe an den SPS gesendet wurde. Der Antrieb sendet dann jedes Mal, wenn eine Befehlsgruppe vom SPS eingeht, eine Antwortgruppe. Wenn die Befehlsgruppe gültig ist und das Antwortgruppentypfeld einen gültigen Antworttyp aufweist, lädt der Antrieb die angeforderten Daten auch in die Antwortgruppe. Die Daten werden bei jedem Abgerufene E/A-Zyklus aktualisiert, bis sich die Befehlsgruppe ändert.
Das Format dieser Gruppe wird in Abschnitt 5.2.1 dargestellt.

Die Antwortgruppe muss im Eingangsspeicher des SPS abgebildet werden. Informationen zur Erzeugung einer Scan-Liste und zur Abbildung des Speichers mit RSNetworx finden Sie im Handbuch Ihres DeviceNet Scanner. Achten Sie darauf, die richtige Steckplatznummer zu verwenden. Für ein Diagramm der Abbildung siehe den vorangegangenen Abschnitt.
Wenn Sie nur einen einzigen SERVOSTAR in der SPS-Scan-Liste haben und sich der Scanner in Einsteckplatz 1 befindet, kann Ihre Abbildung der Eingangsdatei etwa so aussehen:

Eingangswort	Beschreibung
Local:1:I.Data	DN Scanner-Eingangsspeicher
Local:1:I.Data[0]	DN Antwortwort 0 (Statusflags, Blocknummer, Antwortauswahl)
Local:1:I.Data[1]	DN Antwortwort 1 (Antwortdaten)
Local:1:I.Data[2-123]	Nicht abgebildeter Bereich des DN Scanner (steht für andere Geräte zur Verfügung)

Sobald die Eingänge abgebildet sind, können Sie die neueste vom SPS erhaltene Antwortgruppe lesen, indem Sie aus der entsprechenden Eingangsdatei lesen. Die Eingangsdateien werden vom SPS bei jedem Scan-Zyklus aktualisiert, bei dem Daten vom Antrieb übertragen werden.

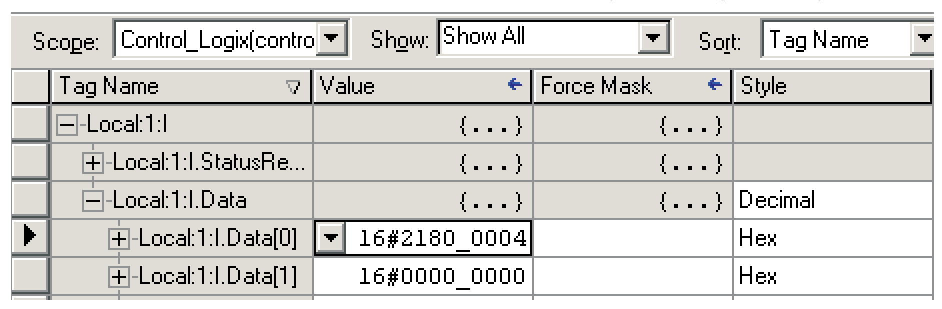
Verwenden Sie testweise die Ausgangsbefehlsgruppe aus dem vorhergegangenen Abschnitt. Die beispielhafte Befehlsgruppe fordert die Istposition an (Antwortgruppentyp = 1). Warten Sie nach dem Schreiben dieser Befehlsgruppe in die Ausgangsdatei, bis der Scan-Zyklus abgeschlossen ist. Lesen Sie dann die Eingangsdatei, um die neueste Antwortgruppe zu erhalten. Die Antwortgruppe sollte etwa wie folgt aussehen; die Istposition des Motors findet sich in Bytes 4-7.

Byte	Funktion	Datenwert (hex)
0	Antwortflags A – Deaktiviert, in Position	0x04
1	Block wird ausgeführt – keine	0x00
2	Antwortflags B – Laden abgeschlossen, keine Fehler	0x80
3	Achseninstanz, Antwortgruppe 1 – Istposition	0x21
4	Istposition – niederwertiges Wort niederwertiges Byte	0x00
5	Istposition – niederwertiges Wort höherwertiges Byte	0x00
6	Istposition – höherwertiges Wort niederwertiges Byte	0x00
7	Istposition – höherwertiges Wort höherwertiges Byte	0x00

Die Abbildung dieser Gruppe im SPS-Speicher sollte etwa so aussehen wie die folgende Tabelle:

Eingangswort	Datenwert (hex)
Local:1:I.Data[0]	0x2180_0004
Local:1:I.Data[1]	0x0000_0000

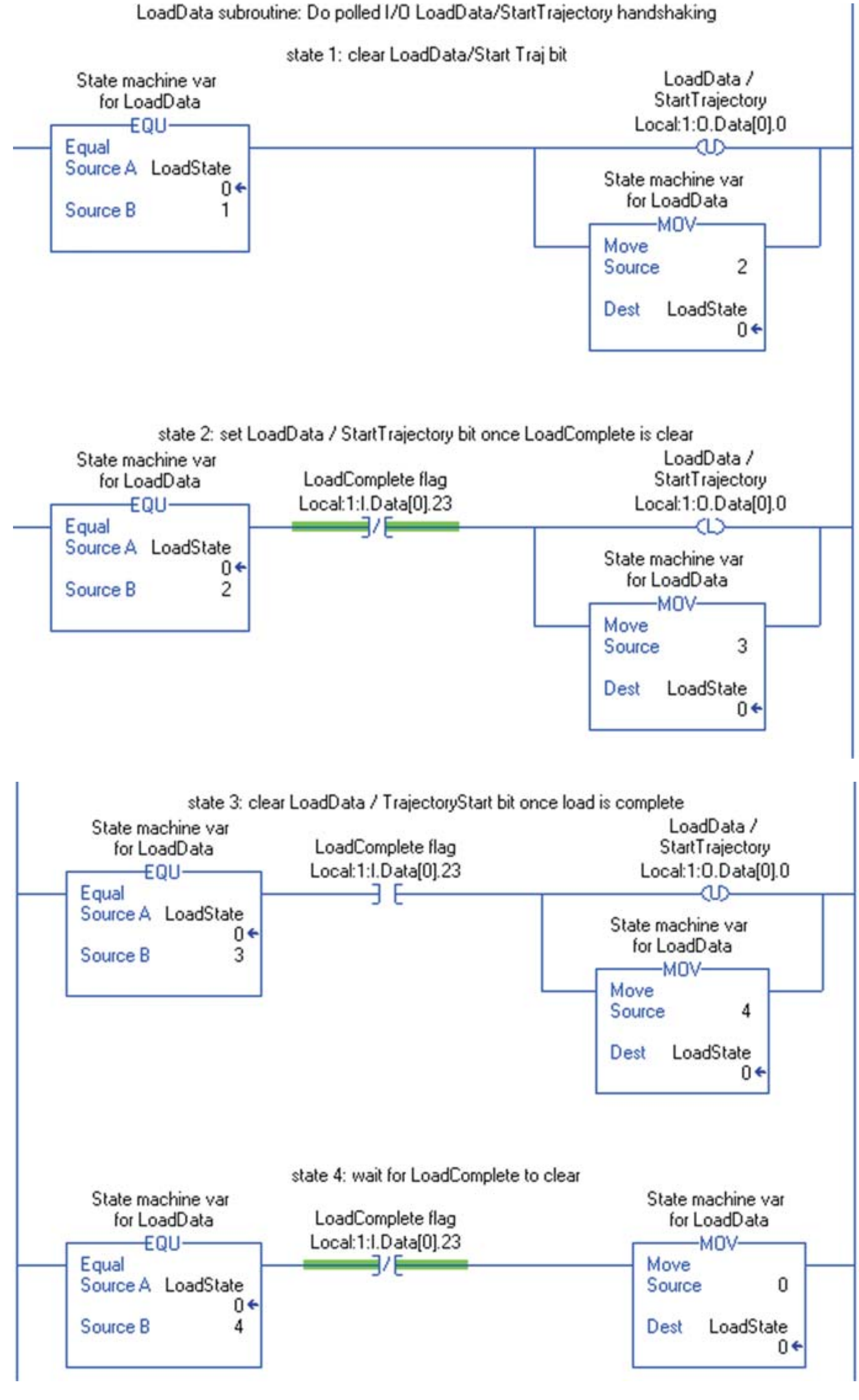
Sie können die Daten einfach in der Bildschirmseite “Programm-Tags” anzeigen.



Der Wert in Local:1:I.Data[1] sollte sich ändern, wenn Sie die Motorwelle von Hand bewegen.

6.1.3.3 Data Handshaking - ControlLogix

Data Handshaking dient zum Übertragen von Datenbefehlen mit Befehlsgruppen. Zum Übertragen eines Befehls an den Antrieb setzen Sie den Befehlstyp und laden Sie Daten in die Datenfelder und wechseln Sie dann das Lade-/Start-Bit auf Hoch. Der Antrieb akzeptiert Daten nur, wenn Laden / Starten von 0 auf 1 wechselt. Wenn die Daten erfolgreich geladen wurden, setzt der Antrieb das Antwortflag “Laden abgeschlossen” auf Hoch. “Laden abgeschlossen” wird vom Antrieb gelöscht, nachdem Laden/Starten vom Regler gelöscht wurde. Für weitere Informationen siehe Abschnitt “Data Handshaking”. Die Unterroutine LoadData kann zur Vereinfachung des Data Handshaking verwendet werden. Die Unterroutine wird im Folgenden gezeigt.

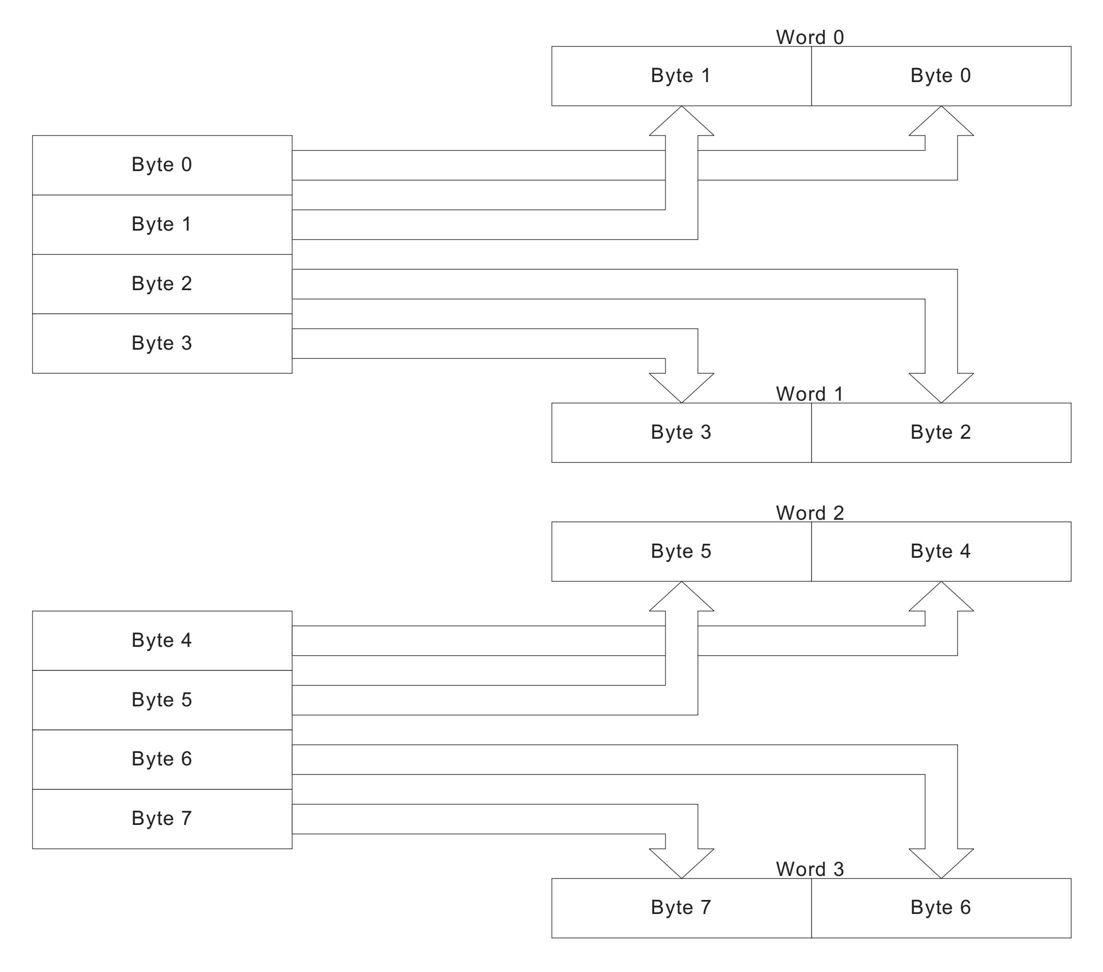


Um LoadData zu verwenden, kopieren Sie zunächst eine entsprechende Befehlsgruppe in Local:1:O.Data[0-1]. Setzen Sie dann LoadState=1 und rufen Sie die Unterroutine LoadData auf, bis LoadState auf 0 zurückgesetzt wurde.

6.1.3.4 Senden von Befehlsgruppen – SLC500

Der SERVOSTAR kann von Allen Bradleys SPSs der SLC -5/0X-Reihe mit DeviceNet Scannern (1756-DNB) unter Verwendung von Gruppenmeldungen gesteuert werden. Befehlsgruppen werden vom SPS an den Antrieb gesendet, um die Bewegung zu steuern. Das Format der Befehlsgruppe wird in Abschnitt 5.1.1 dargestellt.

Für die Kommunikation mit SERVOSTAR-Antrieben über Gruppen muss der Antrieb zunächst in der SPS-Scan-Liste abgebildet werden. (Das Abbildungsverfahren wird in diesem Dokument nicht beschrieben.) Zudem muss die Befehlsgruppe im Ausgabespeicher des SPS abgebildet werden. Informationen zur Erzeugung einer Scan-Liste und zur Abbildung des Eingangs- und Ausgangsspeichers mit RSNetworx finden Sie im Handbuch Ihres DeviceNet Scanner. Achten Sie darauf, die richtige Steckplatznummer zu verwenden. Der S600-Befehl und die Antwortgruppen sind jeweils acht Byte lang und werden als je vier 32-Bit-Wörter im SPS abgebildet.



Wenn Sie nur einen einzigen SERVOSTAR in der SPS-Scan-Liste haben und sich der Scanner in Einsteckplatz 1 befindet, kann Ihre Abbildung der Ausgangsdatei etwa so aussehen:

Ausgangswort	Beschreibung
O:1.0	Steuerwort DN Scanner
O:1.0/0	Ausgänge des DN Scanner aktivieren
O:1.1	DN Befehlswort 0 (Steuerflags, Blocknummer)
O:1.2	DN Befehlswort 1 (Befehl, Antwort)
O:1.3	DN Befehlswort 2 (Daten “least significant word - LSW”)
O:1.4	DN Befehlswort 3 (Daten “most significant word - MSW”)
O:1.5-31	Nicht abgebildeter Bereich des DN Scanner (steht für andere Geräte zur Verfügung)

Sobald die Ausgänge abgebildet sind, können Sie die an den Antrieb zu sendende Befehlsgruppe modifizieren, indem Sie in die entsprechende Ausgangsdatei schreiben. Die neuen Daten werden beim nächsten und jedem folgenden Scanzyklus übertragen, bis die Ausgangsdatei wieder modifiziert wird.

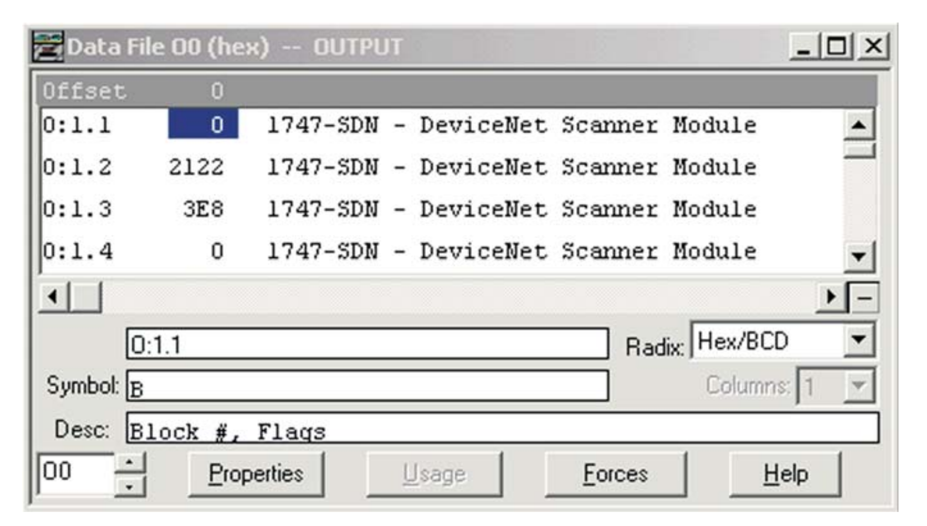
Setzen Sie die Zielgeschwindigkeit testweise auf 1000. Der Wert kann über das Terminal überprüft werden, indem Sie Befehl O_V eingeben. Ändern Sie die SPS-Ausgangsbefehlsgruppe (Scanner-Ausgangsdaten) wie folgt:

Byte	Funktion	Datenwert (hex)
0	Befehlsflags – Deaktivieren	0x00
1	Blocknummer	0x00
2	Achseninstanz, Befehlsgruppe 2 – Zielgeschwindigkeit	0x22
3	Achseninstanz, Antwortgruppe 1 – Istposition	0x21
4	Zielgeschwindigkeit – niederwertiges Wort niederwertiges Byte	0xE8
5	Zielgeschwindigkeit – niederwertiges Wort höherwertiges Byte	0x03
6	Zielgeschwindigkeit – niederwertiges Wort niederwertiges Byte	0x00
7	Zielgeschwindigkeit – höherwertiges Wort höherwertiges Byte	0x00

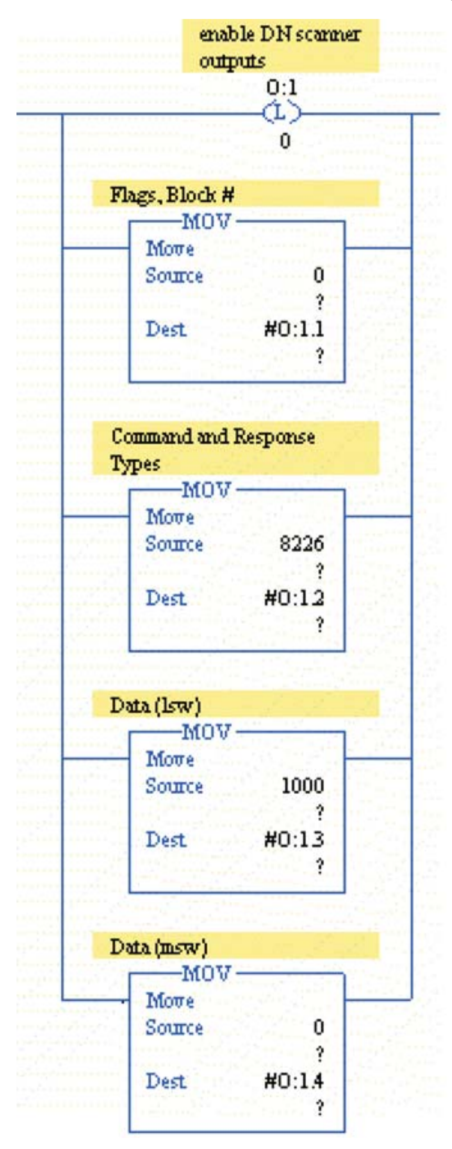
Die Abbildung dieser Gruppe im SPS-Speicher sollte etwa so aussehen wie die folgende Tabelle:

Ausgangswort	Datenwert (hex)
O:1.1	0000
O:1.2	2122
O:1.3	03E8
O:1.4	0000

Sie können die Daten testweise direkt von der Bildschirmseite “Programm-Tags” modifizieren.



Ebenso kann ein Leiterprogramm die gewünschten Werte in die Ausgangswörter laden (beachten Sie, dass die Daten hier in Dezimalen angegeben sind).



Der Befehl und die Daten werden nun bei jedem Scanzyklus an den Antrieb übertragen, doch das Data Handshaking-Protokoll muss zum Laden der Daten verwendet werden. Warten Sie, bis die Befehlsgruppe übertragen wurde. Setzen Sie dann das Lade-Bit auf Hoch, indem Sie 0x0001 in Befehlswort 0 (Ausgangswort O:1.1) schreiben. Verwenden Sie nun das serielle Terminal zum Lesen des Werts von O_V – dieser sollte gleich 1000 sein.

Flags, Block #
MOV
Move
Source 1
?
Dest #O:1.1
?

6.1.3.5 Lesen von Antwortgruppen – SLC500

Eine Antwortgruppe ist eine Abgerufene E/A-Meldung, die vom Antrieb als Antwort auf eine Befehlsgruppe an den SPS gesendet wurde. Der Antrieb sendet dann jedes Mal, wenn eine Befehlsgruppe vom SPS eingeht, eine Antwortgruppe. Wenn die Befehlsgruppe gültig ist und das Antwortgruppentypfeld einen gültigen Antworttyp aufweist, lädt der Antrieb die angeforderten Daten auch in die Antwortgruppe. Die Daten werden bei jedem Abgerufene E/A-Zyklus aktualisiert, bis sich die Befehlsgruppe ändert.

Das Format dieser Gruppe wird in Abschnitt 5.2.1 dargestellt.

Die Antwortgruppe muss im Eingangsspeicher des SPS abgebildet werden. Informationen zur Erzeugung einer Scan-Liste und zur Abbildung des Speichers mit RSNetworx finden Sie im Handbuch Ihres DeviceNet Scanner. Achten Sie darauf, die richtige Steckplatznummer zu verwenden. Für ein Diagramm der Abbildung siehe den vorangegangenen Abschnitt.
Wenn Sie nur einen einzigen SERVOSTAR in der SPS-Scan-Liste haben und sich der Scanner in Einsteckplatz 1 befindet, kann Ihre Abbildung der Eingangsdatei etwa so aussehen:

Eingangswort	Beschreibung
I:1.0	Statuswort DN Scanner
I:1.1	DN Antwortwort 0 (Steuerflags, Blocknummer)
I:1.2	DN Antwortwort 1 (Befehl, Antwort)
I:1.3	DN Antwortwort 2 (Daten “least significant word - LSW”)
I:1.4	DN Antwortwort 3 (Daten “most significant word - MSW”)
I:1.5-31	Nicht abgebildeter Bereich des DN Scanner (steht für andere Geräte zur Verfügung)

Sobald die Eingänge abgebildet sind, können Sie die neueste vom SPS erhaltene Antwortgruppe lesen, indem Sie aus der entsprechenden Eingangsdatei lesen. Die Eingangsdateien werden vom SPS bei jedem Scan-Zyklus aktualisiert, bei dem Daten vom Antrieb übertragen werden.

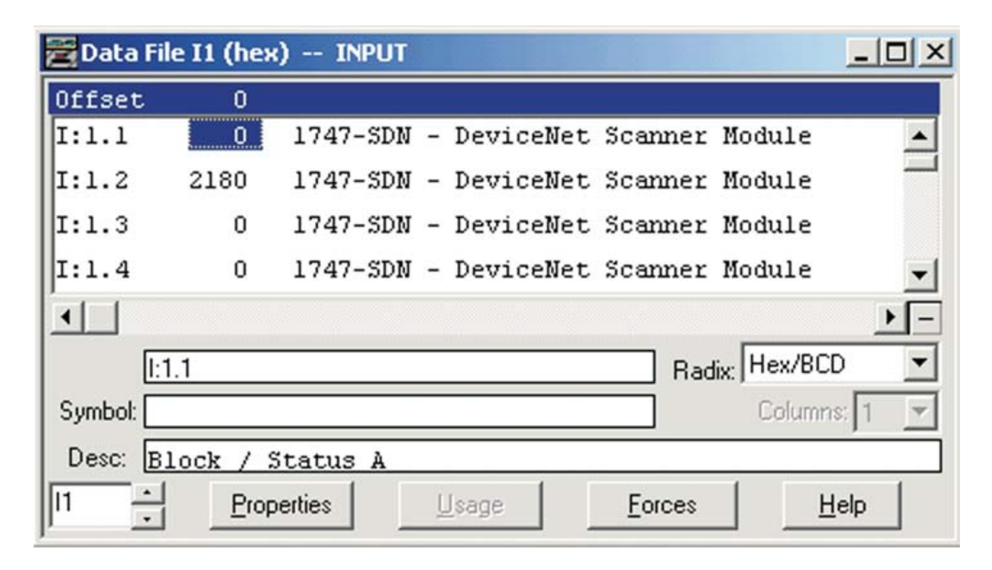
Verwenden Sie testweise die Ausgangsbefehlsgruppe aus dem vorhergegangenen Abschnitt. Die beispielhafte Befehlsgruppe fordert die Istposition an (Antwortgruppentyp = 1). Warten Sie nach dem Schreiben dieser Befehlsgruppe in die Ausgangsdatei, bis der Scan-Zyklus abgeschlossen ist. Lesen Sie dann die Eingangsdatei, um die neueste Antwortgruppe zu erhalten. Die Antwortgruppe sollte etwa wie folgt aussehen; die Istposition des Motors findet sich in Bytes 4-7.

Byte	Funktion	Datenwert (hex)
0	Antwortflags A – Deaktiviert, in Position	0x04
1	Block wird ausgeführt – keine	0x00
2	Antwortflags B – Laden abgeschlossen, keine Fehler	0x80
3	Achseninstanz 1, Antwortgruppe 1 – Istposition	0x21
4	Istposition – niederwertiges Wort niederwertiges Byte	0x00
5	Istposition – niederwertiges Wort höherwertiges Byte	0x00
6	Istposition – höherwertiges Wort niederwertiges Byte	0x00
7	Istposition – höherwertiges Wort höherwertiges Byte	0x00

Die Abbildung dieser Gruppe im SPS-Speicher sollte etwa so aussehen wie die folgende Tabelle:

Eingangswort	Datenwert (hex)
I:1.1	0000
I:1.2	2180
I:1.3	0000
I:1.4	0000

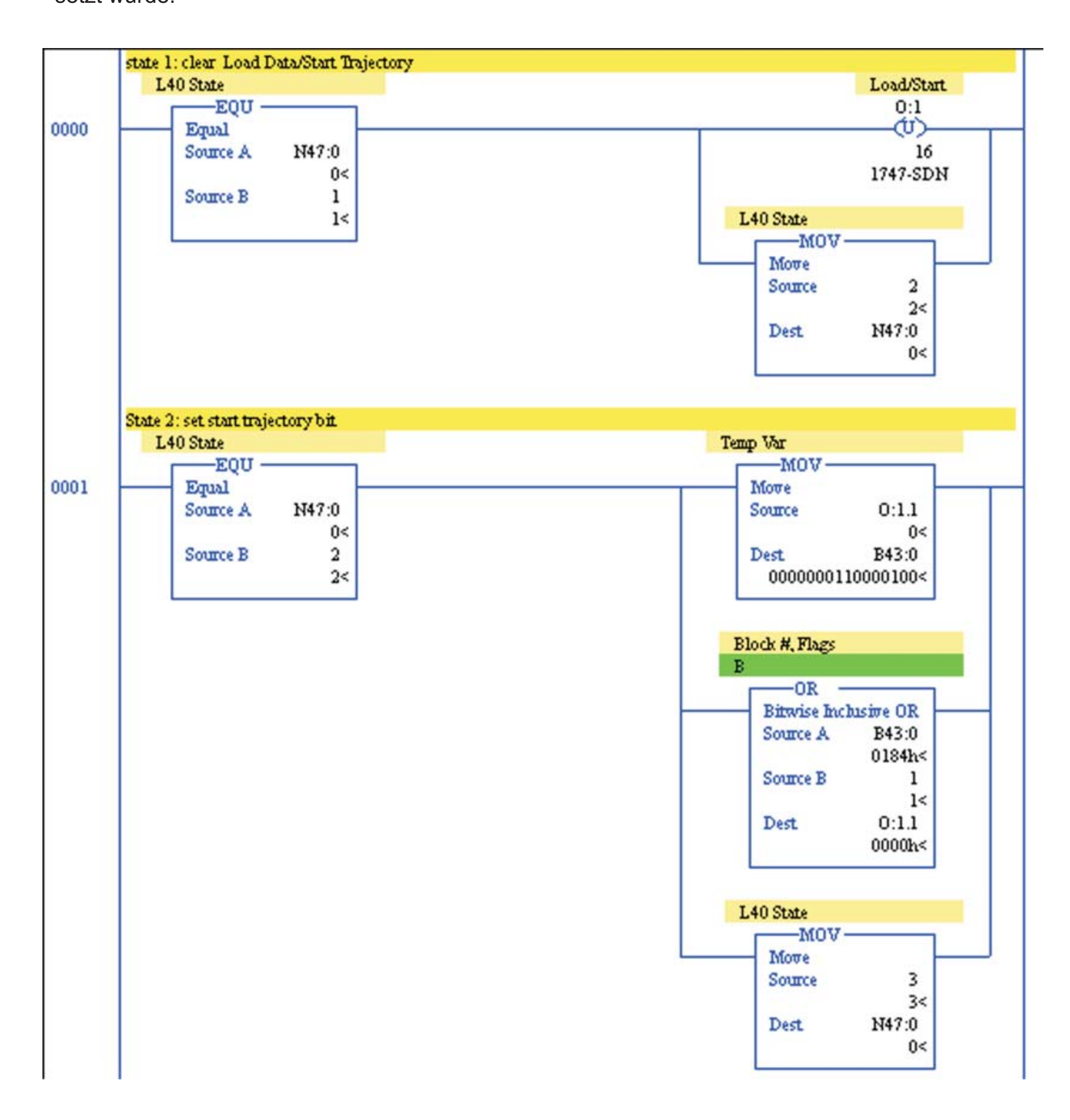
Sie können die Daten einfach in der Bildschirmseite “Programm-Tags” anzeigen.

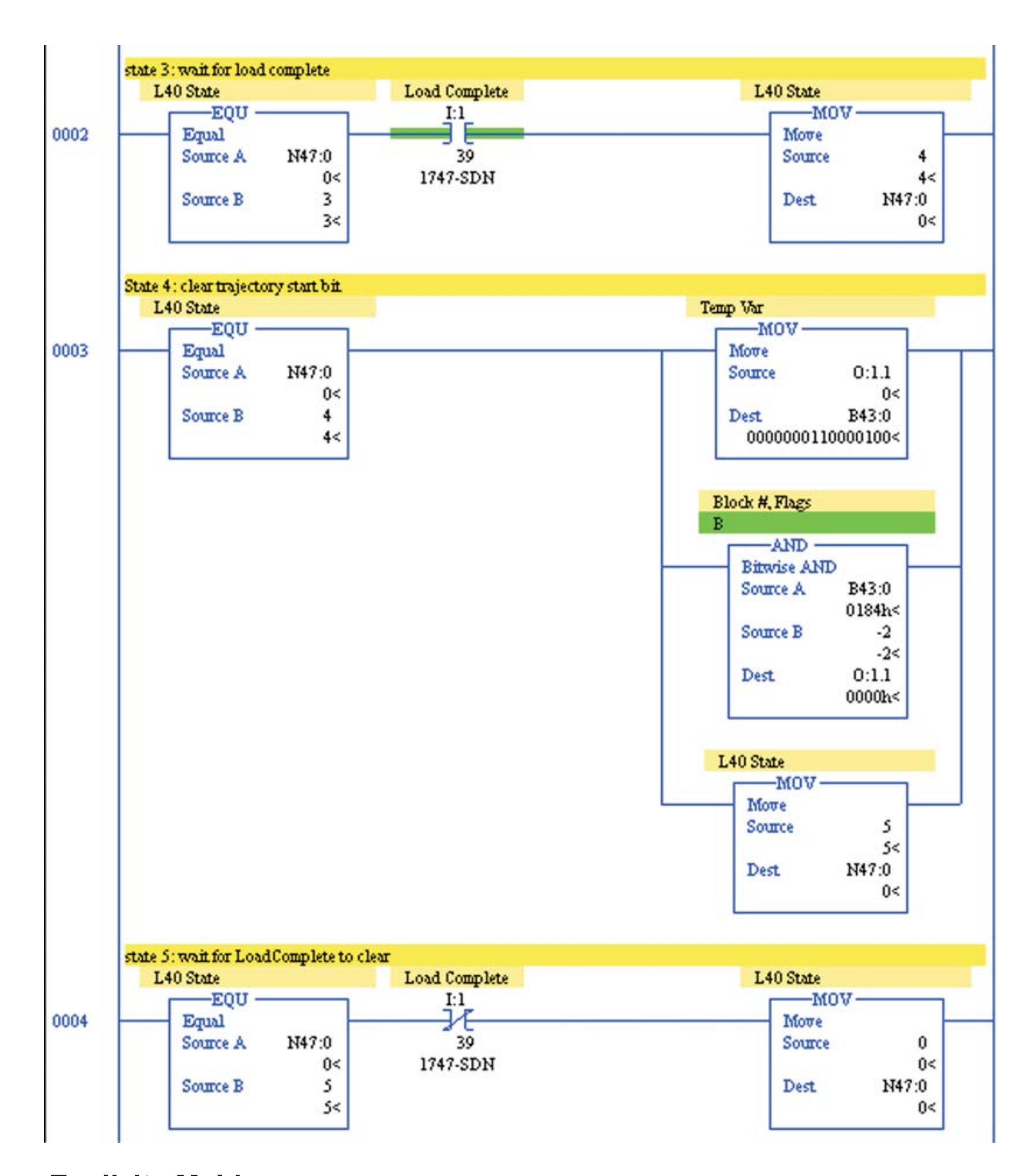


6.1.3.6 Data Handshaking - SLC500

Data Handshaking dient zum Übertragen von Datenbefehlen mit Befehlsgruppen. Zum Übertragen eines Befehls an den Antrieb setzen Sie den Befehlstyp und laden Sie Daten in die Datenfelder und wechseln Sie dann das Lade-/Start-Bit auf Hoch. Der Antrieb akzeptiert Daten nur, wenn Laden/Starten von 0 auf 1 wechselt. Wenn die Daten erfolgreich geladen wurden, setzt der Antrieb das Antwortflag “Laden abgeschlossen” auf Hoch. “Laden abgeschlossen” wird vom Antrieb gelöscht, nachdem Laden/Starten vom Regler gelöscht wurde. Für weitere Informationen siehe Abschnitt “Data Handshaking”.
Die Unterroutine LoadData kann zur Vereinfachung des Data Handshaking verwendet werden. Die Unterroutine wird im Folgenden gezeigt.

Um LoadDaten zu verwenden, kopieren Sie zunächst eine entsprechende Befehlsgruppe in I:1.1-4. Setzen Sie dann N47:0=1 und rufen Sie die Unterroutine LoadData auf, bis N47:0 auf 0 zurückgesetzt wurde.





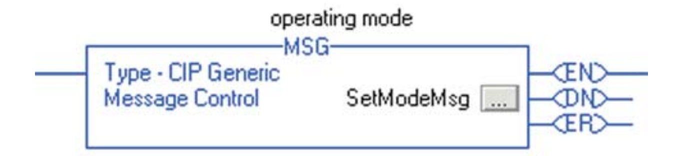
6.1.4 Explizite Meldungen

Eine ausführliche Beschreibung von expliziten Meldungen finden Sie in den folgenden Abschnitten: “Typische Verwendung von expliziten Meldungen und Gruppenmeldungen”, “Explizite Meldungen”," Unterstützte Dienste". In diesen Abschnitten finden Sie Informationen über die Struktur und die Verwendung von expliziten Meldungen.

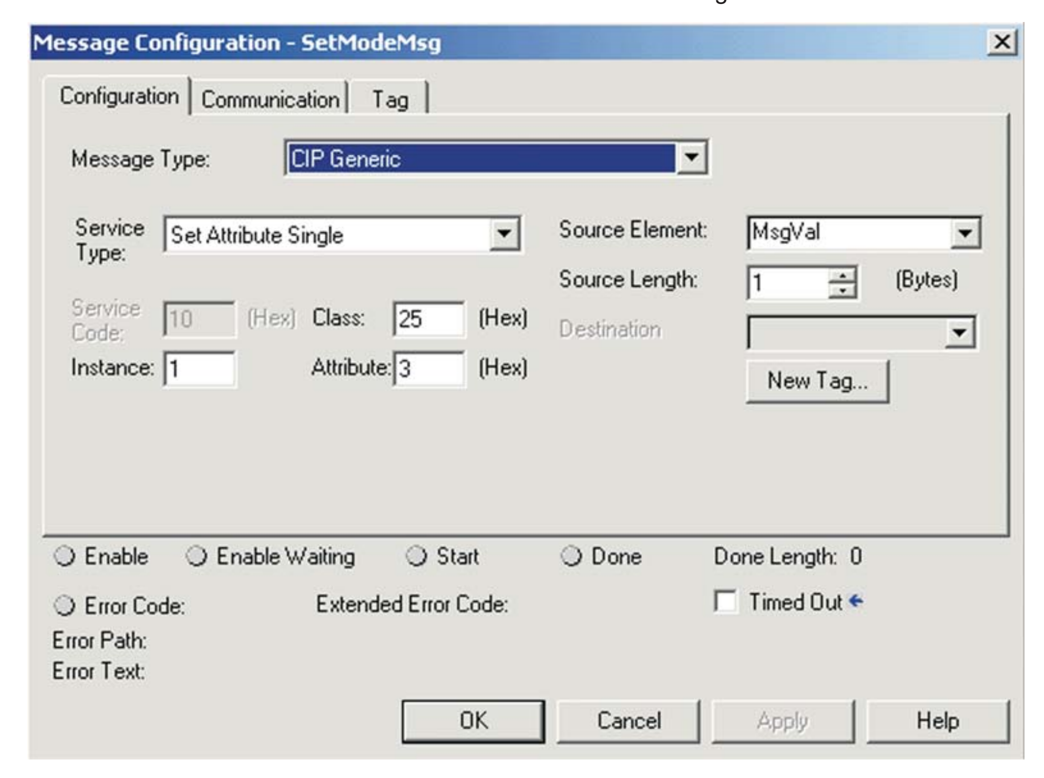
Anders als bei Abgerufene E/A-Gruppenmeldungen werden explizite Meldungen nur dann gesendet, wenn diese explizit vom SPS-Programm angefordert werden. Der SERVOSTAR beantwortet jede explizite Meldung mit Erfolg oder Fehlermeldung.

6.1.4.1 Explizite Meldungen und ControlLogix

Verwenden Sie die ControlLogix Meldungsanweisung MSG zum Übertragen einer expliziten Meldung an den SERVOSTAR.



Das im MSG-Block (SetModeMsg in der Beispielabbildung) verwendete Tag hat Typ MESSAGE und kann durch Klicken auf die Schaltfläche "..." im MSG-Block konfiguriert werden.



Der Meldungstyp für DeviceNet ist "CIP Generic". Im Abschnitt "Unterstützte Dienste" finden Sie eine Beschreibung der im SERVOSTAR verfügbaren Dienste. Die am häufigsten verwendeten Dienste sind "Set Attribute Single" und "Get Attribute Single". Wählen Sie die Werte für Klasse, Instanz und Attribut für die festzulegenden bzw. zu holenden Parameter. Diese Werte werden im Abschnitt "Explizite Meldungen" bereitgestellt. Die Beispielabbildung verwendet Klasse 0x25 – Lageregler-Objekt, Instanz 1 (einzige gültige Instanz für das Lageregler-Objekt), Attribut 3 (OpMode).

Setzen Sie für einen "Set Attribute Single"-Befehl das Quellelement auf eine Variable mit dem zu übertragenden Wert und die Quellenlänge auf den Byte-Zähler des Parameters. Setzen Sie für einen "Get Attribute Single"-Befehl das Ziel auf die Variable, in der der Parameterwert gespeichert werden soll.

Klicken Sie dann auf die Registerkarte "Communication" und geben Sie den Pfad auf den SERVOSTAR ein. Klicken Sie auf "Browse", um Ihren Scanner zu suchen. Das zweite Element des Pfades macht normalerweise 2 aus und bezieht sich auf die externe DeviceNet-Verbindung. Mehr Informationen dazu entnehmen Sie bitte Ihrem Scanner-Handbuch. Das dritte Element ist die Knotenadresse (MACID) des SERVOSTAR.

Im Fall eines Kommunikationsfehlers wird der ER-Ausgang vom MSG-Block gesetzt und der Fehlercode wird auf der unteren Hälfte des Meldungskonfigurationsbildschirms ausgegeben. Für weitere Informationen siehe Abschnitt "Fehlermeldungen".

6.1.4.2 Explizite Meldungen und SLC500

Mit einem SLC500-Prozessor werden explizite Meldungen übertragen, indem sie in die M0-Datei geschrieben werden. Der Erhalt erfolgt durch Lesen aus der M1-Datei. Wenn sich der Scanner in Einsteckplatz 1 befindet, ist die Struktur der expliziten Meldungsanforderung ein 32-Byte-Bereich in M0:1.224-255 und die Struktur der expliziten Meldungsantwort ein 32-Byte-Bereich in M1:1.224-255. Weitere Informationen entnehmen Sie bitte Ihrem Scanner-Handbuch.

Erstellen Sie für dieses Beispiel einen 32-Byte-Ausgangspuffer und einen 32-Byte-Eingangspuffer. Explizite Meldungsanforderungen sind in den Ausgangspuffer eingebaut und werden dann in die M0-Datei kopiert. Die Antwortmeldung in M1 wird zur weiteren Verarbeitung in den Eingangsspeicher kopiert.

6.1.4.2.1 Struktur von expliziten Meldungsanforderungen bei SLC500

Byte 1	Byte 0	SPS-Ausgangspuffer	M0-Speicher
TX_ID = 1	TX_CMD = 1,4	WORD 0	WORD 224
PORT= 0	MSG_SIZE (in Byte)	WORD 1	WORD 225
SERVICE	MAC ID (Knotenadresse)	WORD 2	WORD 226
KLASSE		WORD 3	WORD 227
INSTANZ		WORD 4	WORD 228
ATTRIBUT		WORD 5	WORD 229
LOWER DATA WORD		WORD 6	WORD 230
UPPER DATA WORD		WORD 7	WORD 231

TX_ID Die Transaktions-ID ist ein Index für die explizite Meldungswarteschlange des Scanners. Der Scanner unterstützt Warteschlangen für Meldungen an mehrere Geräte und sendet diese, sobald dies möglich ist.
Für unsere Zwecke in diesem Beispiel verwenden wir immer Transaktions-ID 1.

TX_CMD Der Transaktionsbefehl zum Ausführen auf dem durch TX_ID festgelegten Befehl.
01 Explizite Meldung senden.
04 Antwortpuffer löschen
(vor dem Senden einer neuen Meldung mit derselben TX_ID erforderlich).

PORT 0 – Kanal A (typische Auswahl)
1 – Kanal B

MSG_SIZE Größe aller Daten nach MAC ID in Byte (Wort 3-7). Für eine Meldung, die einen Booleeschen Parameter (1 Byte) festlegt, ist die Meldungsgröße 7, mit jeweils 2 Bytes pro Klasse, Instanz und Attribut plus 1 Byte für den Datenwert.

SERVICE Der auszuführende DeviceNet-Dienst. Für weitere Informationen siehe Abschnitt “Unterstützte Dienste”.
0x0E – Get.
0x10 – Set.

MAC ID Die Adresse des SERVOSTAR, wie durch die beiden MACID-Schalter festgelegt
Wenn die Schalter auf 25 eingestellt sind (die Schalter werden als dezimal gelesen), dann wird dieser Wert auf 19 hex. gesetzt.

KLASSE DeviceNet-Klasse, auf die zugegriffen werden soll. Beispiele:
Objekt “Parameter” – 0x0F
Lageregler-Überwachung – 0x24
Lageregler-Objekt– 0x25

INSTANZ DeviceNet-Instanznummer. Beispiele:
Immer 0x01 für das Lageregler-Objekt.
Parameternummer (DPR-Nummer in ascii-Referenz) für Parameter-Objekt.
Portnummer für analoge und digitale Eingabe/Ausgabe.

ATTRIBUT Die Attributnummer des Attributs, auf das zugegriffen wird (“Set” oder “Get”)

LOWER/UPPER DATA WORD Der Datenwert für einen Set-Dienst.

6.1.4.2.2 Struktur von expliziten Meldungsantworten bei SLC500

Byte 1	Byte 0	SPS-Eingangspuffer	M1 Speicher
TX_ID	TX_STATUS	WORD 0	WORD 224
PORT= 0	MSG_SIZE (in Byte)	WORD 1	WORD 225
SERVICE	MAC ID (in Byte)	WORD 2	WORD 226
DATA		WORD 3 - 31	WORD 227 - 255

TX_ID Transaktions-ID. Entspricht der TX_ID in der Anforderungsmeldung.

TX_STATUS Der Transaktionsstatus für den durch TX_ID festgelegten Transaktionsblock.
0 – Block ignorieren (leer)
1 – Transaktion erfolgreich abgeschlossen
2-15 – Scannerfehler (siehe Scannerdokumentation)

PORT 0 – Kanal A (typische Auswahl)
1 – Kanal B

MSG_SIZE Größe aller Daten nach MAC ID (dem Datenfeld) in Byte.

SERVICE Spiegelt den Servicecode von der Befehlsmeldung, Einstellung des höherwertigen Bit für eine Antwort.
0x1E – Get.
0x90 - Set.
0x94 – DeviceNet-Fehler. Der Fehlercode folgt im Abschnitt “Daten”.

MAC ID Die DeviceNet-ID des SERVOSTAR wie durch die beiden MACID-Schalter festgelegt

DATEN Antwortdaten (Länge in Byte wie mit MSG_SIZE angegeben)

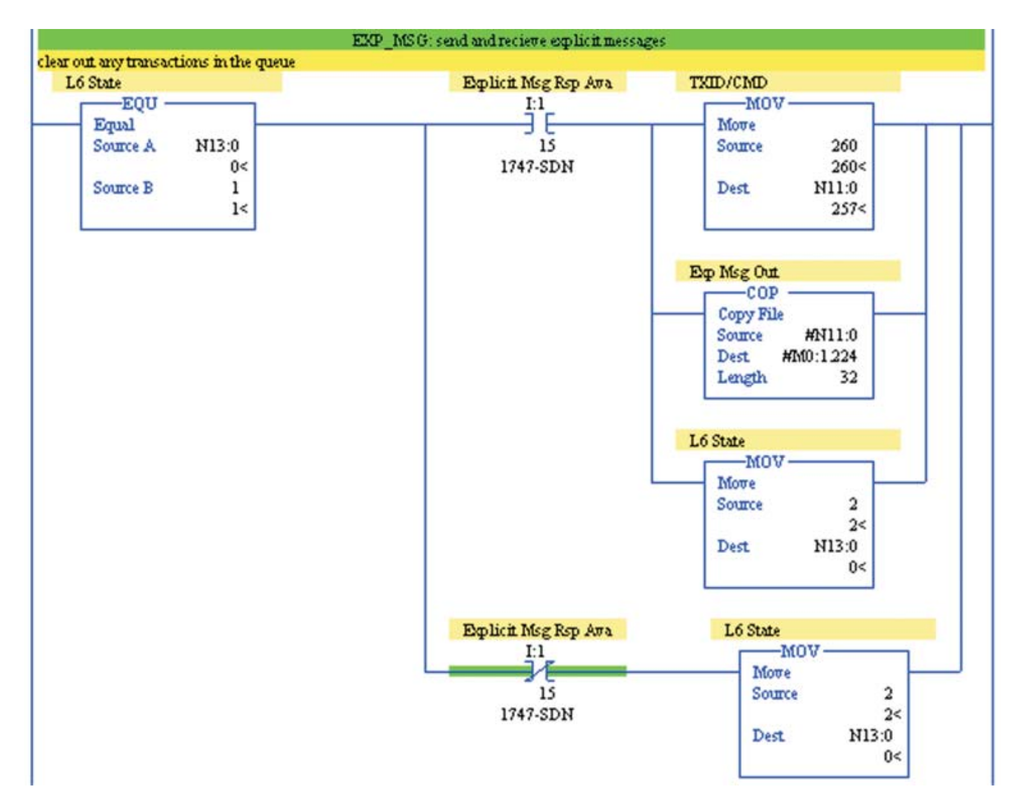
6.1.4.2.3 Explizite Meldungsfolge bei SLC500

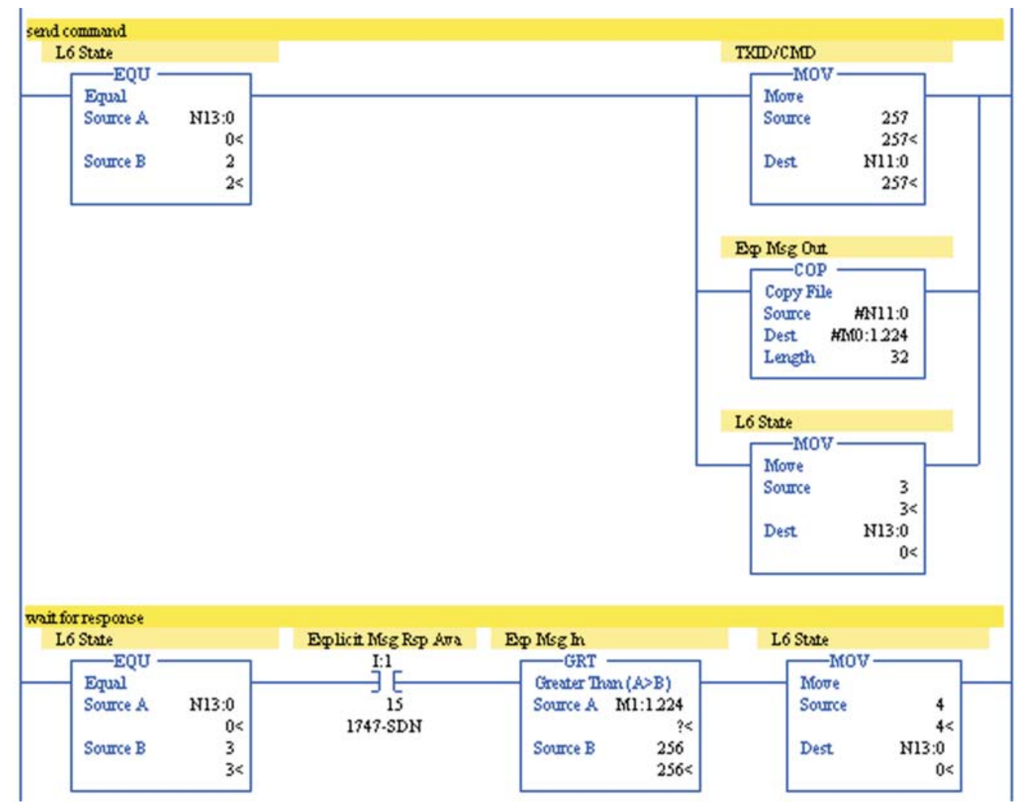
1. Erstellen Sie 32-Byte-Anforderungs- und Antwortpuffer. Das Beispiel verwendet N11 für einen Anforderungspuffer (Ausgang) und N12 für einen Antwortpuffer (Eingang).
2. Löschen Sie den Scanner-Transaktionsblock, den Sie verwenden möchten (durch das Feld TX_ID ausgewählt), indem Sie TX_CMD 0x04 laden (den Antwortpuffer löschen). Um etwa Transaktionsblock 1 zu löschen, setzen Sie TX_ID=0x01 und TX_CMD=0x04 im Ausgangspuffer (WORT 0 = 0x0104), und kopieren Sie dann den Anforderungspuffer in M0:1.224-255.
3. Erstellen Sie im Ausgangspuffer eine explizite Meldungsanforderung. Setzen Sie TX_ID=0x01, um Transaktionsblock 1 zu verwenden, und setzen Sie TX_CMD=0x01 (überträgt explizite Meldung). Verwenden Sie die Kopieranweisung (COP), um die Daten in M0:1.224-255 zu kopieren.
4.. Warten Sie, bis das Explizite Meldungsantwort verfügbar-Bit auf 1 wechselt und damit anzeigt, dass eine explizite Antwortmeldung empfangen wurde. Das Flag ist Bit 1 des 1747-SDN Moduls Status Register (normalerweise in Wort 0 der Eingangsdatei abgebildet, so dass das Bit I:1/155 ist.)
5.. Verwenden Sie die Anweisung zum Kopieren der Datei (COP), um die Daten von M0:1.224-255 in Antwortspuffer6 zu kopieren.
6.. Prüfen Sie das TX_ID-Feld, um sicherzustellen, dass es der in der Anforderungsmeldung festgelegten TX_ID entspricht. Testen Sie den TX_STATUS-Wert auf einen Fehler (1 = Erfolg). Testen Sie den SERVICE-Wert auf einen DeviceNet-Fehlercode (0x94 zeigt einen Fehler an).
7. Löschen Sie den Scanner-Transaktionsblock, indem Sie TX_CMD=0x04 in den Anforderungspuffer laden und den Puffer in M0:1.224-255 kopieren. Nach dem Laden der Daten sollte das Explizite Meldungsantwort verfügbar-Bit auf 0 wechseln.
8. Senden Sie die nächste explizite Anforderungsmeldung, indem Sie bei Schritt 3 fortsetzen.

Hinweis: Bei den SLC500 SPSs dauert es manchmal bis zu 2 Sekunden, bis ein expliziter Meldungsbefehl verarbeitet wird. Diese Verzögerung wird in keinster Weise vom SERVOSTAR gesteuert, da dieser in weniger als 5 ms auf den expliziten Meldungsbefehl reagiert.

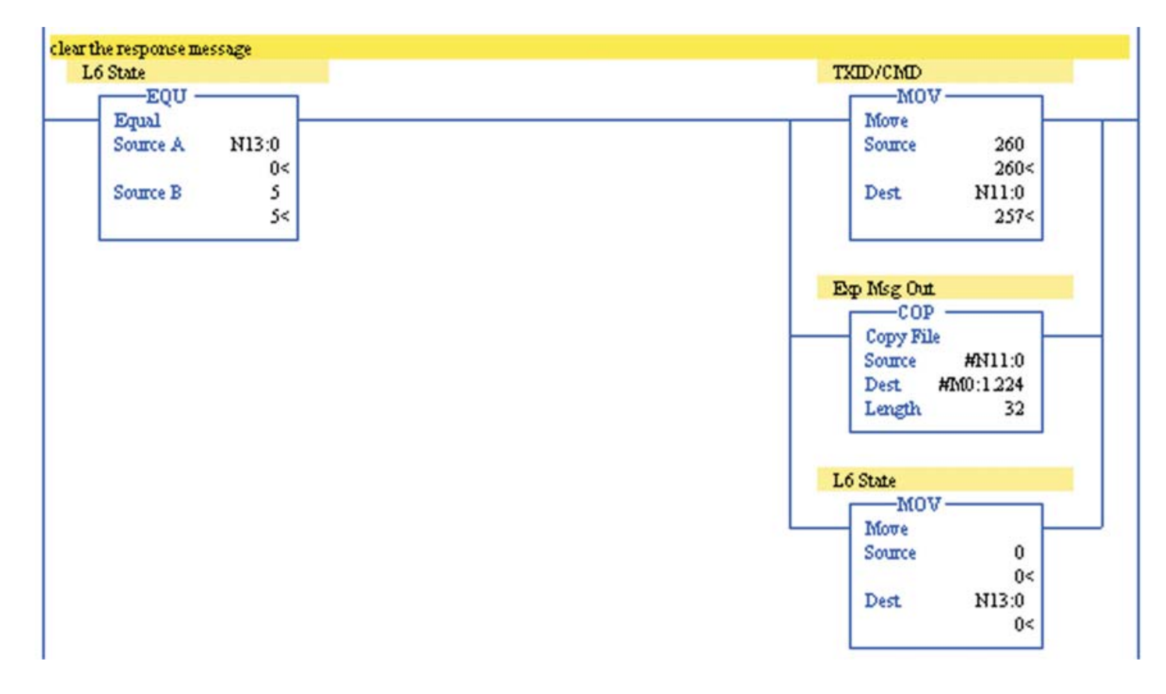
6.1.4.2.4 Beispielcode für Explicit Messaging bei SLC500

Das SLC500-Beispielprogramm enthält die Unterroutine EXP_MSG, um die meisten expliziten Meldungssequenzen verarbeiten zu können. Um die Unterroutine zu verwenden, erstellen Sie einfach eine explizite Anforderungsmeldung im Anforderungspuffer N11. Ignorieren Sie WORT 0 (TX_ID und TX_CMD), da dies von der Unterroutine gesteuert wird. Setzen Sie N13:0 = 1, um die EXP_MSG Maschine zu starten. N13:0 wird auf 0 zurückgesetzt, nachdem die Antwortmeldung in den Antwortpuffer geladen und die Unterroutine beendet wurde.









6.1.5 Beispiel 1: Simple Move

Dieses Beispiel führt einen einfachen Vorgang bestehend aus folgenden Aufgaben durch:

1) Auswählen des Positionsmodus
2) Aktivieren des SERVOSTAR
3) Referenzfahrt
4) Bewegen um eine Umdrehung
5) Warten, bis die Bewegung abgeschlossen ist
6) Setzen des digitalen Ausgangs 1 auf EIN
7) Verzögern um 2 Sekunden
8) Setzen des digitalen Ausgangs 1 auf AUS
9) Zurückbewegen um eine halbe Umdrehung

6.1.5.1 Serielle Befehlssequenz

Dieses Beispiel kann mit folgenden Befehlen manuell vom seriellen Terminalfenster ausgeführt werden.

Befehl	Beschreibung
OPMODE 8	Wechseln in Positionsmodus
EN	Aktivieren des SERVOSTAR
MH	Referenzfahrt
ACCR 10	Beschleunigungszeit = 10ms
DECR 10	Zeit für Geschwindigkeitsabnahme = 10ms
O_V 60	Zielgeschwindigkeit = 60 min-1
O_P 1000	Zielposition = 1000 Schritte (1 Umdrehung
O_C 10240 O_C2 256	Befehlswort = 0x2800 - Benutzereinheiten für Position und Geschwindigkeit; absolute Bewegung; Interpretation O_C2 sekundäres Befehlswort = 0x100 - Verwenden Sie ACCR und DECR für diesen Fahrsatz
MOVE 0	Ausführen von Bewegungsaufgabe 0 (warten Sie, bis die Bewegung abgeschlossen ist)
O1 1	Setzen des digitalen Ausgangs 1 auf Hoch (warten Sie 2 Sekunden auf die gewünschte Verzögerung)
O1 0	Setzen des digitalen Ausgangs 1 auf Niedrig
O_P 500	Setzen der Zielposition auf 500 Schritte (eine halbe Umdrehung)
MOVE 0	Ausführen von Bewegungsaufgabe 0

Testen Sie das Beispiel manuell, ehe Sie fortfahren, um sicherzustellen, dass der SERVOSTAR korrekt konfiguriert ist.

6.1.5.2 DeviceNet-Befehlssequenz

DeviceNet-Befehl	Serielle Terminal-Verifizierung
Setzen Sie OpMode = Position_Mode Explizite Anforderung: Service = 0x10, Klasse = 0x25, Instanz=0x01, Attribut = 0x03, Daten = 0x00	OPMODE 8
Aktivieren Sie den Antrieb Befehlsgruppe: Setzen Sie das “Enable”-Flag = 1 (Byte 0, Bit 7) Wort 0 = 0x2020_0080 Wort 1 = 0x0000_0000	READY 1
Move Home Explizite Anforderung: Service = 0x10, Klasse = 0x0F, Instanz = 141 (MH-Befehl in der ascii-Referenz), Attribut = 0x01, Daten = 0x01	DRVSTAT bit 0x2000 is set
Wait Until Homed – Read Drive Status Word Explizite Anforderung: Service 0x0E, Klasse 0x25, Instanz = 0x01, Attribut = 0x66	Home has completed
Setzen Sie Acceleration Rate = 10 Befehlsgruppe: Verwenden Sie Befehl 0x03 Wort 0 = 0x2023_0080 Wort 1 = 0x0000_000A Wechseln Sie von Bit 0 Hoch auf Laden (Data Handshaking)	ACCR 10
Setzen Sie Deceleration Rate = 10 Befehlsgruppe: Verwenden Sie Befehl 0x04 Wort 0 = 0x2024_0080 Wort 1 = 0x0000_000A Wechseln Sie von Bit 0 Hoch auf Laden (Data Handshaking)	DECR 10
Setzen Sie Target Velocity = 60 Befehlsgruppe: Verwenden Sie Befehl 0x02 Wort 0 = 0x2022_0080 Wort 1 = 0x0000_003C Wechseln Sie von Bit 0 Hoch auf Laden (Data Handshaking)	O_V 60
Setzen Sie Target Position = 1000 Befehlsgruppe: Verwenden Sie Befehl 0x01 Wort 0 = 0x2021_0080 Wort 1 = 0x0000_03E8 Wechseln Sie von Bit 0 Hoch auf Laden und starten Sie die Bewegung (Data Handshaking)	O_P 1000 O_C 10240 = 0x2800 (Benutzereinheiten, 1 Umdr.)
Beachten Sie das Flag Response Assembly – In Motion (Byte 0, Bit 0). Warten Sie, bis dieses Bit auf “Niedrig” wechselt (was anzeigt, dass die Bewegung beendet wurde).	Das Flag “In Motion” sollte “Hoch” sein, sobald der Antrieb in Bewegung ist.
Setzen Sie Digital Output 1 Ein Explizite Anforderung Service = 0x10, Klasse = 0x09, Instanz = 0x01 (für Ausgang1), Attribut = 0x03, Daten = 0x01	O1 1
Verzögern Sie um 2 Sekunden	(Setzen Sie den SPS-Timer auf eine Verzögerung von 2 Sekunden)
Setzen Sie Digital Output 1 Aus Explizite Anforderung Service = 0x10, Klasse = 0x09, Instanz = 0x01 (für Ausgang 1), Attribut = 0x03, Daten = 0x00	O1 0
Setzen Sie Target Position = 500 Befehlsgruppe: Verwenden Sie Befehl 0x01 Wort 0 = 0x2021_0080 Wort 1 = 0x0000_01F4 Wechseln Sie von Bit 0 Hoch auf Laden und starten Sie die Bewegung (Data Handshaking)	O_P 500 O_C 10240 = 0x2800 (Benutzereinheiten, eine halbe Umdrehung)

6.1.5.3 ControlLogix Programm

Ein ControlLogix-Programm namens S600_Example_1.ACD zur Implementierung dieses Beispiels ist auf der Website verfügbar. Das Beispiel geht davon aus, dass ein DeviceNet-Scanner in Einsteckplatz 1 installiert ist und dass ein SERVOSTAR auf Adresse 1 in den niederwertigsten Wörtern des Scanners abgebildet ist.
Um mit der Beispielssequenz zu beginnen, konfigurieren Sie den Antrieb wie im Abschnitt “Antriebseinrichtung für die Beispiele” beschrieben, laden Sie das Programm auf den Prozessor herunter, gehen Sie in den Ausführungsmodus und setzen Sie Ex1State=1.

6.1.5.4 SLC500 Programm

Ein SCL500-Programm namens S600_Example_1.RSS zur Implementierung dieses Beispiels ist auf der Website verfügbar. Das Beispiel geht davon aus, dass ein DeviceNet-Scanner in Einsteckplatz 1 installiert ist und dass ein SERVOSTAR auf Adresse 1 in den niederwertigsten Wörtern des Scanners abgebildet ist.
Um mit der Beispielssequenz zu beginnen, konfigurieren Sie den Antrieb wie im Abschnitt “Antriebseinrichtung für die Beispiele” beschrieben, laden Sie das Programm auf den Prozessor herunter, gehen Sie in den Ausführungsmodus und setzen Sie Ex1State (N16:0) = 1.

6.2 Schaltereinstellungen für die Baudrate

Für SERVOSTAR-Antriebe kann die Baudrate auf 0 (125 KBaud), 1 (250 KBd) oder 2 (500 KBd) gesetzt werden. Wenn der Schalter auf einen höheren Wert als 2 gesetzt wird, ist die Baudrate über den Terminal-Parameter DNBAUD und über DeviceNet konfigurierbar. Wird der Schalter von 0 auf 2 gesetzt, kann die Baudrate nicht mit DNBAUD oder DeviceNet gesteuert werden.

6.3 Konfiguration der MAC ID-Schalter

Werte zwischen 0 und 63 sind zulässig. Wenn diese Schalter auf einen höheren Wert als 63 gesetzt werden, ist die MAC über den Terminal-Parameter DNMACID und über DeviceNet konfigurierbar. Wenn die Schalter von 0 auf 63 gesetzt werden, kann die Baudrate nicht mit DNBAUD oder DeviceNet gesteuert werden.

6.4 Netzwerk LED

Die Netzwerk LED Zeigt den Gerätezustand und den Status im DeviceNet-Netzwerk an.

Gerätezustand	LED	Details
Nicht eingeschaltet / nicht online	aus	Das Gerät ist nicht online.Entweder ist das NEtzwerk abgeschaltet oder das Gerät hat den Dup_MAC_ID-Test noch nicht abgeschlossen.
Online, nicht verbunden	blinkt grün	Das Gerät läuft im normalen Zustand, ist online, und die Verbindungen sind nicht im etablierten Zustand. Das Gerät hat den Dup_MAC_ID-Test bestanden und ist online, aber die Verbindungen zu anderen Knoten sind nicht hergestellt. Dieses Gerät ist keinem Master zugewiesen.
Online, verbunden	grün	Das Gerät läuft im normalen Zustand, ist online, und die Verbindungen sind im etablierten Zustand. Das Gerät ist einem Master zugewiesen.
Connection time-out	blinkt rot	Die abgefragte E/A verbindung ist im Zustand "Time-out"
Kritischer Verbindungs-fehler	rot	Das Gerät hat einen Fehler festgestellt, der die Kommunikation mit dem Netzwerk verhindert (z. B. doppelte MAC ID oder BUSOFF).

6.5 Befehlsübersicht

Dieser Anhang bietet Querverweise von den DeviceNet-Meldungen (sowohl Explicit Message Klasse-/Instanz-/Attributzuordnungen und Abgerufene E/A Befehls-/Antwortfelder) zu den seriellen Terminal-Befehlen. Für Informationen über jeden der seriellen Befehle lesen Sie den entsprechenden Abschnitt in der ASCII-Referenz.

6.5.1 Datentypen

Typ	Beschreibung	Länge (byte)	Min	Max
BOOL	Boolean	1		
SINT	Short Integer	1	-128	127
BYTE	Bit-Zeichenkette – 8 Bits	1		
USINT	Unsigned Short Integer	1	0	255
INT	Integer	2	-32768	32767
UINT	Unsigned Integer	2	0	655351
WORD	Bit-Zeichenkette -16 Bits	2		
DINT	Double Integer	4	-2^{31}	$2^{31}-1$
UDINT	Unsigned Double Integer	4	0	$2^{32}-1$
DWORD	Bit-Zeichenkette -32 Bits	4		
LINT	Long Integer	8	-2^{63}	$2^{63}-1$
ULINT	Unsigned Long Integer	8	0	$2^{64}-1$
LWORD	Bit-Zeichenkette -64 Bits	8		
EPATH	DeviceNet-Pfadsegmente	variabel		

6.5.2 Explizite Meldungen

Services: Reset=0x05, Get=0x0E, Set=0x10
Klassen: 0x01=Objekt identifizieren, 0x24=Lageregler-Überwachung, 0x25=Objekt Lageregler , 0x26=Blockfolgesteuerung, 0x27=Befehlsblock, 0x0F=Parameterobjekt

Name	Kl.	Inst.	Attr.	Di.	Daten	Bemerkung
Objektklasse Identität						
Reset	0x01	0x01	0x00	R	keine	COLDSTART. Auf 1 setzen für Neustart
Default	0x01	0x01	0x01	R	keine	RSTVAR, SAVE, COLDSTART. Auf 1 setzen für Defaultparameter laden und Neustart
Seriennummer	0x01	0x01	0x06	G	UDINT	SERIALNO
Objektklasse Lageregler Überwachung						
Allgemeiner Fehler	0x24	0x01	0x05	G	BOOL	Ist 1 bei allen Verstärkerfehlern und bei Warnmeldungen n3, n8 und n9.
Fehler löschen	0x24	0x01	0x65	G/S	BOOL	CLRFAULT. auf 1 setzen, um Fehler zu löschen.
Fehlercode	0x24	0x01	0x64	G	DINT	ERRCODE. Fehlercode im Hex-Format
Objektklasse Lageregler						
Mode	0x25	0x01	0x03	G/S	USINT	0=Position (OPMODE 8). 1= Geschwindigkeit (OPMODE 0). 2=Drehmoment (OPMODE 2).
Zielposition	0x25	0x01	0x06	G/S	DINT	O_P. Setzt O_C Bits 0x2800 und O_C2 Bits 0x100 so dass Einheiten SI sind und ACCR/DECR O_ACC1/O_DEC1 übersteuern.
Zielgeschw.	0x25	0x01	0x07	G/S	DINT	O_V. Einheit wird von Verstärker vorgegeben.
Beschleunigung	0x25	0x01	0x08	G/S	DINT	Geschwindigkeitsmodus: ACC. Positionsmodus: ACCR. Einheiten werden von ACCUNIT/PGEARI vorgegeben
Verzögerung	0x25	0x01	0x09	G/S	DINT	Geschwindigkeitsmodus: DEC Positionsmodus: DECR. Einheiten werden von ACCUNIT/PGEARI vorgegeben
Positionsflag	0x25	0x01	0x0a	G/S	BOOL	O_C Bit 0. 0->ABS, 1->INCR.
Trajektorie Start/Ende	0x25	0x01	0x0b	G/S	BOOL	Auf 1 setzen, um Bewegung zu starten, bleibt auf 1 bis Bewegung abgeschlossen.
In Position	0x25	0x01	0x0c	G	BOOL	INPOS
Istposition	0x25	0x01	0x0d	G/S	DINT	Get: Istposition auslesen (PFB). Set: Position neu definieren. ROFFS=X, NREF=0, MH
Istgeschwindigkeit	0x25	0x01	0x0e	G	DINT	abs(PV)
Enable	0x25	0x01	0x11	G/S	BOOL	EN
Kontrollierter Stopp	0x25	0x01	0x14	G/S	BOOL	STOP mit Verzögerungsrate von DECR. 0 als Antwort.
Sofortiger Stopp	0x25	0x01	0x15	G/S	BOOL	Nothalt, Verzögerungsrate von DECSTOP. Antwort 0.
Tippgeschw.	0x25	0x01	0x16	G/S	DINT	Positiver Wert, Richtung abh. von Attribut23 / J, Einheiten durch Verstärker, verwendet im Geschwindigkeitmodus
Richtung	0x25	0x01	0x17	G/S	BOOL	Get: aktuelle Bewegungsrichtung (nicht das Flag). S: Richtungs-Flag für J. 1->Pos Richtung. 0->neg Richtung

Name	Kl.	Inst.	Attr.	Di.	Daten	Bemerkung
Referenzrichtung	0x25	0x01	0x18	G/S	BOOL	DIR (invers). 1=pos, 0=neg (0 -> CW ist positiv), nur bei disabeltem Verstärker einstellbar, dann speichern und Neustart
Drehmoment	0x25	0x01	0x19	G/S	DINT	Drehmomentbefehl im Drehmomentmodus, interne Einheit, 3280 = Maximaldrehmoment
Parameter Speichern	0x25	0x01	0x65	G/S	BOOL	SAVE. Auf 1 setzen, um Parameter im EPROM zu speichern, erhält 0 als Antwort
Antriebsstatus	0x25	0x01	0x66	G	DINT	DRVSTAT
Trajektoriestatus	0x25	0x01	0x67	G	DINT	TRJSTAT
Objektlkasse Blockfolgesteuerung						
Block	0x26	0x01	0x01	G/S	USINT	1-255. Nummer des auszuführenden Blocks
Block ausführen	0x26	0x01	0x02	G/S	BOOL	startet die Blockausführung, bleibt auf 1, bis die Bewegung abgeschlossen ist. MOVE x.
Aktueller Block	0x26	0x01	0x03	G/S	USINT	Nummer des aktuell ausgeführten Blocks. 0 im Tippbetrieb. TASKNUM.
Blockfehler	0x26	0x01	0x04	G/S	BOOL	Blockfehler, wird beim Auslesen des Blockfehlercodes zurückgesetzt
Blockfehlercode	0x26	0x01	0x05	G/S	USINT	0 = kein Fehler, 1 = ungültiger oder leerer Block, 2 = Wartezeit überschritten, 3 = Ausführungsfehler
Zähler	0x26	0x01	0x06	G/S	DINT	(positiver) Zähler für Blockschleifen. M LOOPCNT.
Objektklasse Befehlsblock (für alle Befehle)						
Blockbefehl	0x27	0x01-0xff	0x01	G/S	USINT	auszuführendes Kommando. Der Wert von Blockbefehl bestimmt das Format von Attribut 3-7. Der Blockbefehl wird in de unteren 2 Byte von O_C2 gespeichert. Für Fahrsätze wird bit 0x100 in O_C2 gesetzt, so dass ACCR und DECR als Beschleunigungs-/Verzögerungsraten verwendet werden. Einstellen des Blockbefehls ändert auch O_C - setzt 0x800 (erweiterter Fahrauftrag) für die meisten Fahraufträge und 0x2800 für Standardfahrsätze.
Blockverknüpfungs nummer	0x27	0x01-0xff	0x02	G/S	USINT	O_FN - Instanznummer des nächsten auszuführenden Blocks, nachdem dieser Block erledigt ist. 0 bedeutet keine Folgeaufgabe.
Objektklasse Befehlsblock (für Befehl 01 "Attribut ändern")						
Attribut ändern – Legt den Wert eines für DeviceNet zugreifbaren Attributs fest. Einstellen des Blockbefehls (Attribut 1) = Attribut ändern setzt auch O_C Bit 0x800 und O_C2=6.						
Zielklasse	0x27	0x01-0xff	0x03	G/S	USINT	Klasse, auf die zugegriffen werden soll (z.B. 0x25 für Objekt Lageregler) Wird im höherwertigen Byte von O_ACC1 gespeichert.
Zielinstanz	0x27	0x01-0xff	0x04	G/S	USINT	Instanz, auf die zugegriffen werden soll. Wird in O_DEC1 gespeichert.
Attributnummer	0x27	0x01-0xff	0x05	G/S	USINT	Attribut, auf das zugegriffen werden soll (einstellbar). Wird im unteren Byte von O_ACC1 gespeichert.
Attributdaten	0x27	0x01-0xff	0x06	G/S	DINT	Neue Attributdaten. Wird in O_P gespeichert.
Objektklasse Befehlsblock (für Befehl 02 "Wartezeit gleich")						
Warten bis Parameter-Wert – Verzögert, bis ein für DeviceNet zugreifbares Attribut einem gewünschten Wert entspricht. Einstellen des Blockbefehls (Attribut 1) = 2 setzt auch O_C Bit 0x800 und O_C2=2.						
Zielklasse	0x27	0x01-0xff	0x03	G/S	USINT	Klasse, auf die zugegriffen werden soll (z.B. 0x25 für Objekt Lageregler) Wird im höherwertigen Byte von O_ACC1 gespeichert.
Zielinstanz	0x27	0x01-0xff	0x04	G/S	USINT	Instanz, auf die zugegriffen werden soll. Wird in O_DEC1 gespeichert.
Attributnummer	0x27	0x01-0xff	0x05	G/S	USINT	Attribut, auf das zugegriffen werden soll (einstellbar). Wird im niederwertigen Byte von O_ACC1 gespeichert.
Wartezeit	0x27	0x01-0xff	0x06	G/S	DINT	Maximale Wartezeit in mx. Fehler bei Erreichen der Zeitüberschreitung. 0 = keine Zeitüberschreitung. Wird in O_FT gespeichert.
Daten vergleichen	0x27	0x01-0xff	0x07	G/S	DINT	Wert, auf den gewartet werden soll. Wird in O_P gespeichert.
Objektklasse Befehlsblock (für Befehl 03 "Verknüpfung größer als")						

Name	Kl.	Inst.	Attr.	Di.	Daten	Bemerkung
Größer als Test – Testet den Wert eines für DeviceNet zugreifbaren Attributs und verzweigt zu einem alternativen Block, wenn der Attributwert größer als der Testwert ist. Einstellen des Blockbefehls (Attribut 1) = 3 setzt auch O_C Bit 0x800 und O_C2=3.						
Zielklasse	0x27	0x01-0xff	0x03	G/S	USINT	Klasse, auf die zugegriffen werden soll (z.B. 0x25 für Objekt Lageregler) Wird im höherwertigen Byte von O_ACC1 gespeichert.
Zielinstanz	0x27	0x01-0xff	0x04	G/S	USINT	Instanz, auf die zugegriffen werden soll. Wird in O_DEC1 gespeichert.
Attributnummer	0x27	0x01-0xff	0x05	G/S	USINT	Attribut, auf das zugegriffen werden soll (einstellbar). Wird im niederwertigen Byte von O_ACC1 gespeichert.
Verknüpfungsnummer vergleichen	0x27	0x01-0xff	0x06	G/S	USINT	Block, zu dem im TRUE-Fall verzweigt wird. Wird in O_DEC2 gespeichert.
Daten vergleichen	0x27	0x01-0xff	0x07	G/S	DINT	Wenn das Attribut größer als Daten vergleichen ist, wird die normale Verknüpfung (Attribut 2) ignoriert und zur alternativen Verknüpfung verzweigt (Attribut 6). Wird in O_P gespeichert.
Objektklasse Befehlsblock (für Befehl 04 "Verknüpfung weniger als")						
Kleiner als Test – Testet den Wert eines für DeviceNet zugreifbaren Attributs und verzweigt zu einem alternativen Block, wenn der Attributwert kleiner als der Testwert ist. Einstellen des Blockbefehls (Attribut 1) = 4 setzt auch O_C Bit 0x800 und O_C2=4.						
Zielklasse	0x27	0x01-0xff	0x03	G/S	USINT	Klasse, auf die zugegriffen werden soll (z.B. 0x25 für Objekt Lageregler) Wird im höherwertigen Byte von O_ACC1 gespeichert.
Zielinstanz	0x27	0x01-0xff	0x04	G/S	USINT	Instanz, auf die zugegriffen werden soll. Wird in O_DEC1 gespeichert.
Attributnummer	0x27	0x01-0xff	0x05	G/S	USINT	Attribut, auf das zugegriffen werden soll (einstellbar). Wird im niederwertigen Byte von O_ACC1 gespeichert.
Verknüpfungsnummer vergleichen	0x27	0x01-0xff	0x06	G/S	USINT	Block, zu dem bei “Wahr” verzweigt werden soll. Wird in O_DEC2 gespeichert.
Daten vergleichen	0x27	0x01-0xff	0x07	G/S	DINT	Wenn das Attribut kleiner als Daten vergleichen ist, wird die normale Verknüpfung (Attribut 2) ignoriert und zur alternativen Verknüpfung verzweigt (Attribut 6). Wird in O_P gespeichert.
Objektklasse Befehlsblock (für Befehl 05 "Zähler herunterzählen")						
Zähler herunterzählen – Dieser Block verringert den globalen Zähler im Objekt “Befehlsblockfolgesteuerung”. Für diesen Befehlstyp werden keine weiteren Attribute definiert. Einstellen des Blockbefehls (Attribut 1) = 5 setzt auch O_C Bit 0x800 und O_C2=9.						
Objektklasse Befehlsblock (für Befehl 06 "Verzögerung")						
Verzögerung – Dieser Block bewirkt, dass die Folgesteuerung für eine bestimmte Anzahl von Millisekunden verzögert wird, ehe mit dem nächsten Block fortgefahren wird. Der Block muss eine Blockverknüpfung in Attribut 2 aufweisen. Einstellen des Blockbefehls=6 setzt auch O_C Bits 0x800 und O_C2=1.						
Verzögerung	0x27	0x01-0xff	0x03	G/S	DINT	Verzögerungszeit in ms. Wird in O_FT gespeichert.
Objektklasse Befehlsblock (für Befehl 08 "Trajektorie")						
Trajektorie – Führt eine Positionierungsbewegung aus. Einstellen des Blockbefehls=8 setzt auch O_C Bit 0x3800 (SI Einheiten, erweiterte Aufgabe) und O_C2=0x100 (Verwenden von globaler Beschleunigungs- und Geschwindigkeitsabnahmerate).						
Zileposition	0x27	0x01-0xff	0x03	G/S	DINT	O_P
Zielgeschw.	0x27	0x01-0xff	0x04	G/S	DINT	O_V
Inkrementell	0x27	0x01-0xff	0x05	G/S	BOOL	O_C Bit 0. 0->abs. Bewegung, 1->rel. Bewegung.
Objektklasse Befehlsblock (für Befehl 09 "Geschwindigkeit ändern")						
Geschwindigkeitsänderung – Führt ein Geschwindigkeitsprofil aus. Einstellen des Blockbefehls=9 setzt auch O_C2=0x165 und O_C2 Bits 0x3800. Dieser Blocktyp kann keine Verknüpfung auf einen Folgeblock aufweisen, da das Geschwindigkeitsprofil kein definitives Ende hat.						
Zielgeschw.	0x27	0x01-0xff	0x03	G/S	DINT	O_V
Objektklasse Parameter						
Die Instanznummer im Parameterobjekt entspricht der DPR-Nummer, die in der seriellen ASCII-Terminalbefehlsreferenz für Antriebsparameter angegeben ist. Nur die Parameter 1-255 sind zugreifbar.						
Parameterwert	0x0F	0x01-0xFF	0x01	G/S	Typ	Istwert des Parameters. Der Wert ist schreibgeschützt, wenn Bit 4 von Attr#4 (Deskriptor-ReadOnly-Bit) 1 ist.

Name	Kl.	Inst.	Attr.	Di.	Daten	Bemerkung
Zugriff	0x0F	0x01-0xFF	0x04	G	WORD	Schreibgeschützt, wenn Bit 0x10 gesetzt ist.
Datenlänge	0x0F	0x01-0xFF	0x06	G	USINT	Länge der Daten in Bytes.
Objektklasse Diskreter Einganspunkt						
Wert	0x08	0x01-0x04	0x03	G	BOOL	0=Aus; 1=Ein. Instanz 1-4 —> IN1, IN2, IN3, IN4. Dies sind die digitalen Onboard-Eingänge, die auf Stecker X3 verfügbar sind.
Objektklasse Diskreter Ausgabepunkt						
Wert	0x09	0x01-0x02	0x03	S	BOOL	0=Aus; 1=Ein. Einstellbar nur wenn O1MODE=23 / O2MODE=23, sonst Fehler 0x10 Device-Konflikt. Wird bei jedem Fehler auf 0 zurückgesetzt. Instanz 1,2 —> O1, O2.
Objektklasse Diskreter Eingangspunkt						
Wert	0x0A	0x01-0x02	0x03	G	INT	Spannung am Eingang in Millivolt. Instanz 1,2 —> ANIN1, ANIN2.
Objektklasse Analoger Ausgabepunkt						
Wert	0x0B	0x01-0x02	0x03	S	INT	Instanz 1,2 —> AN1TRIG,AN2TRIG. Setzt ANOUT1/2 = 6 für DeviceNet-Steuerung. Wert ist die Spannung für den Ausgang in Millivolt. Bei ANOUT1/2 6 wird Fehler 0x10 Device-Konflikt ausgegeben.

6.5.3 Abgerufene E/A-Meldungen

Name	Byte	Bit	Wert	Bemerkung
Befehlsgruppe				
Daten laden/Profil starten	0	0	0/1	Zum Laden der Daten in den Antrieb setzen Sie den Befehlstyp und die Datenfelder, dann wechseln Sie dieses Bit 0->1, um Data Handshaking zu initiieren. Wenn der Befehl akzeptiert wird, setzt der Antrieb das Bit Geladene Daten in der Antwortgruppe. Wenn der Befehlstyp der Antriebsart entspricht, beginnt die Bewegung (Positionsbefehl in Positionsmodus, Drehmomentbefehl in Drehmomentmodus, Geschwindigkeitsbefehl in Geschwindigkeitsmodus).
Startblock	0	1	0/1	Wechsel 0->1 zum Ausführen eines Befehlsblocks oder einer Kette. Die Blocknummer ist in Byte 1 der Befehlsgruppe. Ähnlich MH [BlockNumber]
Relativ	0	2	0/1	O_C Bit 0. 0->absolute Position. 1 = relativ. (Nur in Positionsmodus)
Richtung	0	3	0/1	Steuert die Richtung des Motors im Geschwindigkeitsmodus. 1 = vorwärts, 0 = rückwärts. Ändert die Richtung des Motors, wenn das Bit sich ändert, selbst wenn die Bewegung bereits läuft. Gültig nur im Geschwindigkeitsmodus. Setzt das Vorzeichen von VJOG.
Kontrollierter Stopp	0	4	0/1	STOP. Sofortiger kontrollierter Stopp. Verwendet DECR als Rate.
Sofortiger Stopp	0	5	0/1	DECSTOP. Sofortiger schneller Stopp.
Reg Arm	0	6	0/1	Registrierungsarm.
Aktivieren	0	7	0/1	EN. 1 = Antrieb aktivieren. 0 = Deaktivieren und Bewegung stoppen.
Blocknummer	1	1-7	0-255	Blocknummer zum Ausführen auf einer positiven Startblockflanke. MOVE [BlockNumber]
Befehlstyp	2	0-4	0-5	Befehlstypen folgen noch. Setzen Sie das Bit Daten laden/Trajektorie starten, um den Befehl zu laden.
x00 Keine Aktion	2	0-4	0	Es soll nichts getan werden.
x01 Zielposition	2	0-4	1	O_P. Die Bewegung beginnt, wenn dieser Befehl im Positionsmodus geladen wird.
x02 Zielgeschwindigkeit	2	0-4	2	Positionsmodus: O_V. Geschwindigkeitsmodus: VJOG und Bewegung beginnen, wenn dieser Befehl geladen wird.
x03 Beschleunigung	2	0-4	3	Geschwindigkeitsmodus: ACCR. Positionsmodus: O_ACC1. Einheiten gesetzt von ACCUNIT,PGEARI. Schritte/s²
x04 Verzögerung	2	0-4	4	Geschwindigkeitsmodus: DECR. Positionsmodus: O_DEC1. Einheiten gesetzt von ACCUNIT,PGEARI. Schritte/s²
x05 Drehmoment	2	0-4	5	T. Funktioniert nur im Drehmomentmodus.
Befehlsachse	2	5-7	1	Muss immer 1 sein. Jeder andere Wert macht die Befehlsgruppe ungültig.
Antworttyp	3	0-4	0-3,5,0x14	Antworttypen folgen noch. Die Antwortdaten werden in der nächsten Antwortgruppe sein.
x00 Keine Aktion	3	0-4	0	Es soll nichts getan werden. Antwortdaten werden Nullen sein.
x01 Istposition	3	0-4	1	PFB
x02 Sollposition	3	0-4	2	PTARGET
x03 Istgeschwindigkeit	3	0-4	3	abs(PV). Schritte/s. Absoluter Wert der Geschwindigkeit.
x05 Drehmoment	3	0-4	5	I
x14 Gruppenfehler	3	0-4	0x14	Fehlercode in Antwortgruppe: Bytes 4-5=Fehlercode, 6-7=Befehlsbytes 2-3 spiegeln. FEHLERCODES... Weniger als 8 Bytes: x13 ff. Nicht unterstützter Befehl: x08 01. Nicht unterstützte Antwort: x08 02. Nicht unterstützte Befehlsachse: x05 01. Nicht unterstützte Antwortachse: x05 02. Nicht unterstütztes Attribut holen: x14 02. Nicht unterstütztes Attribut setzen: x14 01. Nicht einstellbares Attribut setzen: x0E FF. Ungültigen Wert setzen: x09 FF.

Name	Byte	Bit	Wert	Bemerkung
Antwortachse	3	5-7	1	Muss immer 1 sein. Jeder andere Wert macht die Befehlsgruppe ungültig.
Befehlsdaten	4-7			Daten abhängig vom Befehlstyp. Datenbytes sind in umgekehrter Reihenfolge – das niederwertigste Byte zuerst.
Antwortgruppe				
Profil läuft ab	0	0	0/1	1 = eine Bewegung wurde befohlen und ist noch nicht beendet. DRVSTAT Bit 0x10000
Block in Ausführung	0	1	0/1	1 = ein Block ist in Ausführung. Blocknummer in Byte 1 angegeben.
In Position	0	2	0/1	INPOS. 1 = in Position
Allgemeiner Fehler	0	3	0/1	1 = Alarm. Fehler, Warnungen n3, n8, n9. ERRCODE.
Aktuelle Richtung	0	4	0/1	Aktuelle Richtung. 1 = Vorwärts. V positiv oder negativ.
Referenzfahrt-Flag	0	5	0/1	1=Flag aus, 0=Flag ein. Ebene des Referenzfahrt-Eingangs. DRVSTAT Bit 0x40000
Reg.ebene	0	6	0/1	Registrierungseingangsebene. IN2MODE muss = 26
Aktivieren	0	7	0/1	1 = aktiviert. READY.
Ausführung Blocknummer	1	0-7	0-255	Block derzeit in Ausführung. 0=Keine Blockausführung. TASKNUM.
Fehlereingang	2	0	0/1	1 = Fehlereingang ist aktiv. Verwenden von Notstoppeingängen. Prüfen auf Eingang mit INxMODE=27 und Eingangsebene nieder (aktive Ebene “Nieder” ist ein Fehler).
Positives HW-Limit	2	1	0/1	DRVSTAT Bit 0x200. 1=aktiv.
Negatives HW-Limit	2	2	0/1	DRVSTAT Bit 0x400. 1=aktiv.
Positives SW-Limit	2	3	0/1	DRVSTAT Bit 0x40. 1=aktiv.
Negatives SW-Limit	2	4	0/1	DRVSTAT Bit 0x20. 1=aktiv.
Folgefehler	2	5	0/1	Folgefehler. DRVSTAT Bit 0x04.
Blockfehler	2	6	0/1	Fehler bei der Ausführung eines Blocks. 1 = Fehler. Lesen von Objektattribut #5 “Blockfolgesteuerung“ zum Löschen.
Laden beendet	2	7	0/1	Laden beendet. Befehlsdaten erfolgreich geladen. Zurücksetzen wenn Bit Laden/Starten ”Nieder" ist.
Antworttyp	3	0-4	0-3,5,0x14	Spiegelt Antworttyp von Befehlsgruppe. Siehe obige Beschreibung.
Antwortachse	3	5-7	1	Spiegelt Antwortachse von Befehlsgruppe.
Antwortdaten	4-7			Daten abhängig vom Antworttyp. Datenbytes sind in umgekehrter Reihenfolge – das niederwertigste Byte zuerst.

6.6 Vorgabe Eingangs-/Ausgangskonfiguration

Für den SERVOSTAR gilt folgende Eingangskonfiguration:

O1MODE=23 (DeviceNet-Steuerung des digitalen Ausgangs 1)
O2MODE=23 (DeviceNet-Steuerung des digitalen Ausgangs 2)
ANOUT1=6 (DeviceNet-Steuerung des analogen Ausgangs 1)
ANOUT1=6 (DeviceNet-Steuerung des analogen Ausgangs 1)

6.7 Fehlermeldungen

DeviceNet-Fehlermeldungen werden ausgegeben, wenn eine Befehlsgruppe oder eine explizite Anforderungsmeldung vom Antrieb nicht erfolgreich verarbeitet werden kann. Das ist oft auf eine ungültige Meldung oder einen ungültigen Antriebszustand zurückzuführen.

Der SERVOSTAR überträgt eine Fehlerantwortgruppe als Antwort auf eine fehlerhafte Befehls-gruppe. Bei dieser Antwort wird Befehl/Antwortfehlergruppentyp 0x14 in das Antwortgruppentypfeld und die Fehlercodes in Byte 4-5 geladen. Weitere Informationen finden Sie im Abschnitt “Antwort-gruppe 0x14".

Der SERVOSTAR überträgt eine explizite Fehlerantwortmeldung als Antwort auf eine fehlerhafte explizite Anforderungsmeldung. Diese Antwort hat Servicecode 0x94 und die Fehlercodes in den ersten beiden Datenbytes.

Fehlercode (hex)	zusätzlicher Code (hex)	DeviceNet Fehler
0	FF	NO ERROR
2	FF	RESOURCE_UNAVAILABLE
5	FF	PATH_UNKNOWN
5	1	COMMAND_AXIS_INVALID
5	2	RESPONSE_AXIS_INVALID
8	FF	SERVICE_NOT_SUPP
8	1	COMMAND_NOT_SUPPORTED
8	2	RESPONSE_NOT_SUPPORTED
9	FF	INVALID_ATTRIBUTE_VALUE
B	FF	ALREADY_IN_STATE
C	FF	OBJ_STATE_CONFLICT
D	FF	OBJECT_ALREADY_EXISTS
E	FF	ATTRIBUTE_NOT_SETTABLE
F	FF	ACCESS_DENIED
10	FF	DEVICE_STATE_CONFLICT
11	FF	REPLY_DATA_TOO_LARGE
13	FF	NOT_ENOUGH_DATA
14	FF	ATTRIBUTE_NOT_SUPP
15	FF	TOO_MUCH_DATA
16	FF	OBJECT_DOES_NOT_EXIST
17	FF	FRAGMENTATION_SEQ_ERR
20	FF	INVALID_PARAMETER

6.8 Firmware-Änderungsprotokoll

6.8.1 Firmware Version 5.56

Erstes Release von Firmware mit DeviceNet.

6.8.2 Firmware Version 5.74

- SAVE wird ausgeführt, wenn die Baud-Rate oder die Stationsadresse über DeviceNet festgelegt wird.
- EXTWD kann zum Deaktivieren der DeviceNet-Knotenüberwachung verwendet werden.
- Standardverhalten bei Kommunikationsabbruch wird erkannt, um die Kommunikation automatisch zurückzusetzen.
- Der Speicherdienst des Parameter-Objekts wird unterstützt.
- DeviceNet-Geschwindigkeitsmodus ist nun im Antrieb OPMODE 0 statt 8 implementiert.
- Lageregler-Attribut 8 – Beschleunigung wird nun im Geschwindigkeitsmodus als ACC und im Positionsmodus als ACCR abgebildet.
- Befehlsgruppe 0x03 – Beschleunigung wird nun im Geschwindigkeitsmodus als ACC und im Positionsmodus als ACCR abgebildet.
- Lageregler-Attribut 9 – Geschwindigkeitsabnahme wird nun im Geschwindigkeitsmodus als DEC und im Positionsmodus als DECR abgebildet.
- Befehlsgruppe 0x04 – Geschwindigkeitsabnahme wird nun im Geschwindigkeitsmodus als DEC und im Positionsmodus als DECR abgebildet.
- Lageregler-Attribut 22 – Tippgeschwindigkeit wird nun als J und nicht als VJOG abgebildet.
- Befehlsgruppe 0x02 – Zielgeschwindigkeit wird nun im Geschwindigkeitsmodus als J und nicht als VJOG abgebildet.
- Lageregler-Attribut 40 – Sollauflösung wird jetzt unterstützt.
- Lageregler-Attribut 41 – Motorauflösung wird jetzt unterstützt.
- Das Antwortgruppe aktuelle Richtung-Bit liefert nun die letzte befohlene Position, wenn keine Bewegung im Gange ist.
- Bit 0x1000 in O_C wird nicht gesetzt, wenn das Attribut Zielposition oder Befehlsgruppe empfangen wird.

6.8.3 Firmware Version 5.82

- Das Attribuit Parameter Nummer wurde dem Objekt Parameter hinzugefügt um den Zugriff auf Parameter > DPR 255 zu ermöglichen.
- Wenn die Schalterstellungen für Baudrate und Adresse beim Booten güktig sind, werden die Werte automatisch in DNBAUD und DNADDR gespeichert.

6.8.4 SERVOSTAR 300 Firmware Version 1.0

- SERVOSTAR 600 DeviceNet Code in den SERVOSTAR 300 übertragen
- DeviceNet Funktionen, die vom SERVOSTAR 300 nicht unterstützt werden
  Überwachungsattribut des Lagereglers, Attribut 0x0E Aktive Indexebene
  Überwachungsattribut des Lagereglers, Attribut 0x15 Registrierung aktivieren
  Überwachungsattribut des Lagereglers, Attribut 0x16 Registrierung Eingangsebene
  Objekt Lageregler, Attribut 0x66 Antriebsstatus
  Objekt Lageregler, Attribut 0x18 Referenzrichtung
  Objekt Blockfolgesteuerung, Attribut 0x06 Zähler
  Objekt Befehlsblock, Befehl 0x01 Attribut ändern
  Objekt Befehlsblock, Befehl 0x02 Wartezeit gleich
  Objekt Befehlsblock, Befehl 0x03 Verknüpfung größer als
  Objekt Befehlsblock, Befehl 0x04 Verknüpfung weniger als
  Objekt Befehlsblock, Befehl 0x05 Zähler herunterzählen
  Objekt Befehlsblock, Befehl 0x06 Verzögerung
  Objekt Befehlsblock, Befehl 0x09 Tippbetrieb
  Objekt Identität, Dienst 0x05 Reset, DEFAULT_RESET nicht unterstützt
  Objekt analoger Ausgang
  Befehlsgruppen, Byte 0, Bit 0x40 Registrierung aktivieren
  Node Guarding und EXTWD

DeviceNet Funktionen, die für den SERVOSTAR 300 hinzugefügt wurden
Objekt Befehlsblock, Attribut 0x64 O_C
Objekt Befehlsblock, Attribut 0x65 O_ACC
Objekt Befehlsblock, Attribut 0x66 O_DEC
Objekt Befehlsblock, Attribut 0x67 O_TAB
Objekt Befehlsblock, Attribut 0x68 O_FT

6.9 Index

Diese Seite wurde bewusst leer gelassen.

Vertrieb und Service

Wir wollen Ihnen einen optimalen und schnellen Service bieten. Nehmen Sie daher bitte Kontakt zu der für Sie zuständigen Vertriebsniederlassung auf. Sollten Sie diese nicht kennen, kontaktieren Sie bitte den europäischen oder nordamerikanischen Kundenservice.

Europa

Besuchen Sie die europäische Danaher Motion Website auf www.DanaherMotion.net.
Dort finden Sie die aktuelle Inbetriebnahmesoftware, Applikationshinweise und die neuesten Produkthandbücher.

Danaher Motion Kundenservice - Europa

Internet www.DanaherMotion.net
E-Mail support@danahermotion.net
Tel.: +49(0)203 - 99 79 - 0
Fax: +49(0)203 - 99 79 - 155

Nordamerika

Besuchen Sie die nordamerikanische Danaher Motion Website auf www.DanaherMotion.com.
Dort finden Sie die aktuelle Inbetriebnahmesoftware, Applikationshinweise und die neuesten Produkthandbücher.

Danaher Motion Customer Support North America

Internet www.DanaherMotion.com
E-Mail customer.support@danahermotion.com
Tel.: 1-540-633-3400
Fax: 1-540-639-4162

